

**Reduced Order Modeling for  
Stochastic Prediction and Data Assimilation  
Onboard Autonomous Platforms At Sea**

by

Jacob Peter Heuss

B.S., Purdue University, 2013

Submitted to the Mechanical Engineering Department  
in partial fulfillment of the requirements for the degree of  
Master of Science, Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

Sep 2021

©2021 Jacob P Heuss.

All rights reserved.

The author hereby grants to MIT and WHOI permission to reproduce and to  
distribute publicly paper and electronic copies of this thesis document in whole or in  
part in any medium now known or hereafter created.

Author .....  
Mechanical Engineering Department  
Massachusetts Institute of Technology  
& Woods Hole Oceanographic Institution  
Aug 13, 2021

Certified by .....  
Pierre Lermusiaux  
Professor of Mechanical Engineering and Ocean Science and Engineering  
Associate Department Head for Research and Operations  
Massachusetts Institute of Technology  
Thesis Supervisor

Accepted by .....  
Nicolas Hadjiconstantinou  
Chairman, Committee for Graduate Students  
Massachusetts Institute of Technology

Accepted by .....  
David Ralston  
Chairman, Joint Committee for Applied Ocean Science & Engineering  
Woods Hole Oceanographic Institution



**Reduced Order Modeling for  
Stochastic Prediction and Data Assimilation  
Onboard Autonomous Platforms At Sea**

by

Jacob Peter Heuss

Submitted to the Mechanical Engineering Department  
Massachusetts Institute of Technology  
& Woods Hole Oceanographic Institution  
on Aug 13, 2021, in partial fulfillment of the  
requirements for the degree of  
Master of Science, Mechanical Engineering

**Abstract**

There are many significant challenges for unmanned autonomous platforms at sea including predicting the likely scenarios for the ocean environment, quantifying regional uncertainties, and updating forecasts of the evolving dynamics using their observations. Due to the operational constraints such as onboard power, memory, bandwidth, and space limitations, efficient adaptive reduced order models (ROMs) are needed for onboard predictions. In the first part, several reduced order modeling schemes for regional ocean forecasting onboard autonomous platforms at sea are described, investigated, and evaluated. We find that Dynamic Mode Decomposition (DMD), a data-driven dimensionality reduction algorithm, can be used for accurate predictions for short periods in ocean environments. We evaluate DMD methods for ocean PE simulations by comparing and testing several schemes including domain splitting, adjusting training size, and utilizing 3D inputs. Three new approaches that combine uncertainty with DMD are also investigated and found to produce practical and accurate results, especially if we employ either an ensemble of DMD forecasts or the DMD of an ensemble of forecasts. We also demonstrate some results from projecting / compressing high-fidelity forecasts using schemes such as POD projection and K-SVD for sparse representation due to showing promise for distributing forecasts efficiently to remote vehicles. In the second part, we combine DMD methods with the GMM-DO filter to produce DMD forecasts with Bayesian data assimilation that can quickly and efficiently be computed onboard an autonomous platform. We compare the accuracy of our results to traditional DMD forecasts and DMD with Ensemble Kalman Filter (EnKF) forecast results and show that in Root Mean Square Error (RMSE) sense as well as error field sense, that the DMD with GMM-DO errors are smaller and the errors grow slower in time than the other mentioned schemes. We also showcase the DMD of the ensemble method with GMM-DO. We conclude that due to its accurate and computationally efficient results, it could be readily applied onboard autonomous platforms. Overall, our contributions developed and integrated stochastic DMD forecasts and efficient Bayesian GMM-DO updates of the DMD state and parameters, learning from the limited gappy observation data sets.

Thesis Supervisor: Pierre Lermusiaux

Title: Professor of Mechanical Engineering and Ocean Science and Engineering  
Associate Department Head for Research and Operations  
Massachusetts Institute of Technology

## Acknowledgments

First, I wish to thank the U.S. Navy’s Civilian Institution Program along with the MIT-WHOI Joint Program for providing the funding and resources that made this research and continuing my education possible. I am also grateful to the U.S. Navy for continuing to provide the opportunity and challenge of working alongside some of the brightest people on Earth at two world-class institutions.

I wish to extend my thanks to Professor Pierre Lermusiaux, my research advisor. His mentorship, guidance, and flexibility throughout my time in the Joint Program as a member of the Multidisciplinary, Simulation, Estimation, and Assimilation Systems (MSEAS) laboratory has really helped me develop both personally and professionally. While also a leader of the Mechanical Engineering Department at MIT, with a busy and demanding schedule, he made my research, as well as my personal and professional development, a priority. By empowering me to leverage my Naval experience to gain insight into research topics, he gave me a better understanding of the benefits and challenges of collaboration between academia and the United States Department of Defense.

Thank you to my MSEAS colleagues for the help and friendship that you have provided. The banter we shared on a near daily basis really helped me continue to feel connected to MIT during the COVID-19 pandemic. Although we shared frustrations and troubles, together we continue to learn and develop. Without these brilliant individuals’ assistance and support this research would not have been possible. They always provided their time to provide insight, correction, and stimulation to my research, while completing demanding research and challenging classes of their own. I am truly grateful for their support.

Thank you to all of the MIT-WHOI staff. Thank you to Lisa Cherin-Mayer, Pat Haley, and Chris Mirabito for the assistance and expertise you provide to MSEAS.

PFJL and the MSEAS group are grateful to the Office of Naval Research for support under STTR grant N6833519C0348 (ROMs) to the Massachusetts Institute of Technology (MIT).

Finally, I want to thank my wonderful wife, Carla, and my incredibly smart son, Jeremy. To my wife, you have inspired me since the day we met and continue to give me a reason to be a better person. To my son, you make me proud everyday and I will look back fondly on the opportunity to spend time with you during this challenging time at MIT-WHOI.

THIS PAGE INTENTIONALLY LEFT BLANK

# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Motivation . . . . .	15
1.2	Challenges and Contributions . . . . .	15
1.3	Marine Autonomy . . . . .	17
1.4	Thesis Overview . . . . .	19
<b>2</b>	<b>Reduced Order Models and Dynamic Mode Decomposition</b>	<b>21</b>
2.1	DMD Origins . . . . .	21
2.2	DMD Architecture . . . . .	21
2.3	DMD Methods . . . . .	23
2.3.1	Projected DMD . . . . .	24
2.3.2	Exact DMD . . . . .	25
2.3.3	Compressed DMD . . . . .	26
2.3.4	Total DMD . . . . .	27
2.3.5	Optimized DMD . . . . .	28
2.3.6	Streaming DMD . . . . .	30
2.4	DMD of the Ensemble . . . . .	32
2.5	Reduced Order Projection and Subspace Methods . . . . .	33
2.5.1	POD, PCA, and EOF Projections . . . . .	34
2.5.2	K-SVD and other sparse Projections . . . . .	34
2.6	DMD and Projection Results and Discussions . . . . .	35
2.6.1	DMD Method Predictions . . . . .	36
2.6.2	Splitting the Domain into Regions . . . . .	37
2.6.3	Training Size . . . . .	38

2.6.4	3D Forecasting . . . . .	38
2.6.5	Ensemble Mean and Variance . . . . .	39
2.6.6	DMD of the Ensemble . . . . .	42
2.6.7	Subspace Projections . . . . .	44
2.6.8	Summary . . . . .	45
<b>3</b>	<b>Data Assimilation</b>	<b>47</b>
3.1	Introduction . . . . .	47
3.2	Ensemble Kalman Filter . . . . .	48
3.2.1	Description . . . . .	48
3.2.2	Formulation . . . . .	49
3.2.3	Error Subspace Statistical Estimation . . . . .	50
3.3	GMM-DO Filter . . . . .	51
3.3.1	Gaussian Mixture Models . . . . .	52
3.3.2	Expectation-Maximization Algorithm . . . . .	53
3.3.3	The EM algorithm used with Gaussian Mixture Models . . . . .	56
3.3.4	Bayesian Inference Criterion . . . . .	58
3.3.5	Dynamically Orthogonal Field Equations . . . . .	60
3.4	DMD with GMM-DO Filter Description . . . . .	62
3.5	Summary . . . . .	68
<b>4</b>	<b>DMD with DA Test Cases and Applications</b>	<b>69</b>
4.1	Flow Behind an Island/Cylinder Test Case . . . . .	69
4.1.1	Description of Test Case . . . . .	70
4.1.2	Simulated Flow Snapshots and Measurements . . . . .	75
4.1.3	Ensemble POD Modes . . . . .	75
4.1.4	DMD with DA Results for Flow Behind an Island/Cylinder . . . . .	77
4.1.5	DMD Forecasts . . . . .	77
4.1.6	DMD with DA Results . . . . .	78
4.2	Realistic Ocean Simulation Test Case . . . . .	82
4.2.1	Description of POSYDON-POINT Test Case . . . . .	82
4.2.2	POSYDON-POINT Snapshots . . . . .	82
4.2.3	Ensemble POD Modes . . . . .	83



4.2.4	DMD of the Ensemble results for POSYDON-POINT Inputs . . . . .	83
4.3	Computational Costs . . . . .	85
<b>5</b>	<b>Conclusion and Future Work</b>	<b>89</b>
5.1	Conclusion . . . . .	89
5.2	Future Work . . . . .	90

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Figures

2-1	SST errors of 12 hour forecast for persistence (left), Total DMD applied to entire domain simultaneously (center), Total DMD on shelf/deep water separately (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment. . . . .	37
2-2	Zonal velocity truth (left), errors for the persistence (second from left) and exact DMD forecasts on Sep 16, 2006 19Z, for the AWACS/SW06 experiment, training with either 300 snapshots (second from right) or 800 snapshots (right). . . . .	38
2-3	SST errors of 84 hour forecast for persistence (left) and Exact DMD (right) on Aug 25, 2006 19Z using 199 training snapshots (with 108 DMD modes used) from the AWACS/SW06 experiment. . . . .	39
2-4	SST error from the true SST mean of 12 hour forecast for persistence (left), mean of Total DMD forecasts of all ensembles (second from left), mean of Total DMD forecasts of 41 ensembles (second from right), and Total DMD forecast using mean field (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment. . . . .	41
2-5	Standard deviation fields for true SST, the Total DMD SST forecasts of all ensemble members (PCC of 0.80), the 41 ensemble members Total DMD SST forecasts (PCC of 0.62), and the Total DMD forecast using variance as the input (PCC of 0.82) on August 27, 2018 00Z for the POSYDON-POINT experiment. . . . .	42
2-6	Surface Zonal Velocity errors for August 27, 2018 00Z for the POSYDON-POINT experiment. The reconstructed 12 hour PE forecast projection on past POD modes (left) with a PCC of 0.81 and the persistence forecast (right) with a PCC of 0.26. . .	44
3-1	Parametric (Gaussian) distribution, GMM, and Kernel Density approximation of 20 samples generated from mixture of uniform distributions: $p_X(x) = \frac{1}{2}\mathcal{U}(x; -8, -1) + \frac{1}{2}\mathcal{U}(x; 1, 8)$ where $\mathcal{U}(x; a, b) = \frac{1}{b-a}$ is the continuous uniform probability density function for random variable $X$ . Source [109]. . . . .	53

3-2	DMD with GMM-DO Filter flowchart. . . . .	63
4-1	General setup of domain for flow behind an island/cylinder. . . . .	74
4-2	Example of vorticity field (top plot) and two sensors (bottom plot) for flow behind an island/cylinder at $Re = 200$ . Bottom plot shows the vorticity at sensor one (labeled $s_1$ ) vs. time in a blue solid line and the vorticity at sensor 2 (labeled $s_2$ ) vs. time in an orange dotted line. . . . .	76
4-3	Mean vorticity field (a) and POD singular values (b) for flow behind an island/cylinder with $Re = 200$ . The first 6 POD modes are shown in (c)-(j). . .	76
4-4	Real part of the first two Dominant DMD modes. (a) and (c) are extracted from the 'truth' where as (b) and (d) are extracted from the ensemble. . . .	77
4-5	True vorticity field 0.2 s past the last training snapshot (top plot), DMD error field at 0.2 s past the last training snapshot, and the pattern correlation coefficients (PCCs) for persistence and DMD. . . . .	78
4-6	Same as Fig. 4-5 except 5 s past last training snapshot. . . . .	79
4-7	Same as Fig. 4-5 except 20 s past last training snapshot. . . . .	79
4-8	True velocity field 0.2 s past the last training snapshot (top plot), DMD with EnKF error field at 0.2 s past the last training snapshot (second from top), DMD with EnKF error field at 0.2 s past the last training snapshot (second from bottom), and the pattern correlation coefficients (PCCs) for DMD, DMD with EnKF, and DMD with GMM-DO. . . . .	80
4-9	Same as Fig. 4-8 but at 5 s past the last training snapshot. . . . .	81
4-10	Same as Fig. 4-8 but at 20 s past the last training snapshot. . . . .	81
4-11	Example of sea surface temperature field (top plot left), sea surface temperature field sub-region (top plot right), and two sensors (bottom plot) for POSYDON-POINT Experiment simulation in the Middle Atlantic-New York Bight region on 27 August 2018 [65]. Sensors ( $s_1$ and $s_2$ ) are as labeled and represent sea surface temperature sensing locations from 23 August 2018 to 27 August 2018 for demonstration purposes. . . . .	83

4-12	(a) Full region SST field 27 August 2018 00Z, (b) sub-region SST field 27 August 22018 00Z, (c) Mean SST field of the training snapshots, (d) POD singular values for indicated sub-region from the POSYDON-POINT Experiment simulation in the Middle Atlantic-New York Bight region for 23 Aug 2018 to 27 August 2018. The first 4 POD modes of the sub-region are shown in (e)-(h).	84
4-13	(a) True SST field for the 5 <i>hr</i> forecast, (b) true SST field for the 5 <i>hr</i> forecast, (c) RMSEs for DMD (blue), DMD with EnKF (orange), and DMD with GMM-DO (yellow), (d) DMD SST 5 <i>hr</i> forecast field error, (e) DMD with EnKF SST 5 <i>hr</i> forecast field error, and (f) DMD with GMM-DO SST 5 <i>hr</i> forecast field error.	85
4-14	Same as Fig. 4-13 but for the 10 <i>hr</i> forecast.	86
4-15	Same as Fig. 4-13 but for the 20 <i>hr</i> forecast.	86
4-16	Same as Fig. 4-13 but for the 30 <i>hr</i> forecast.	87
4-17	Same as Fig. 4-13 but for the 40 <i>hr</i> forecast.	87

THIS PAGE INTENTIONALLY LEFT BLANK

# Chapter 1

## Introduction

### 1.1 Motivation

Unmanned and autonomous platforms at sea are playing a larger role throughout the world in scientific and defense communities. In January 2021, the United States Navy Chief of Naval Operations (CNO) , Admiral Michael M. Gilday, released his CNO Navigation Plan 2021. Admiral Gilday makes clear that "Unmanned platforms play a vital role in our future fleet" [35]. The ability of autonomous platforms to store stochastic and/or deterministic forecasts is still limited (despite increasing capabilities).

While several organizations, including the US Navy, produce ensemble forecasts, loading these large high fidelity forecasts directly onto an autonomous vehicle is, at present, not feasible. One alternative to storing directly on the autonomous platform is the use of compression techniques and reduced order model (ROM) techniques that allow for the platform to make its own local forecasts at much reduced storage and computational costs. These ROM techniques, when combined with data assimilation (DA), allow for the platform to operate for much longer periods of time with reliable forecasts that are based on previous high fidelity forecasts, real-time sensed data, and periodic compressed communications from outside platforms (like a reach-back cell, ship, or another nearby autonomous platform).

### 1.2 Challenges and Contributions

There are many significant challenges for unmanned autonomous platforms at sea including predicting the likely scenarios for the ocean environment, quantifying regional uncertainties,

and updating forecasts of the evolving dynamics using their observations [60, 61]. Due to the operational constraints such as onboard power, memory, bandwidth, and space limitations, efficient adaptive reduced order models (ROMs) are needed for onboard predictions. In addition we discuss how Dynamic Mode Decomposition (DMD) [53, and references therein], a data-driven dimensionality reduction algorithm, can be used for accurate predictions for short periods in ocean environments [41]. In this thesis, several reduced order modeling schemes for regional ocean forecasting onboard autonomous platforms at sea are described (Section 2.3), investigated, and evaluated (Section 2.6). We evaluate Dynamic Mode Decomposition (DMD) methods for ocean PE simulations by comparing several schemes including domain splitting, adjusting training size, and utilizing 3D inputs. Three different approaches that combine uncertainty with DMD are also investigated and found to produce practical and accurate results, especially if we employ either an ensemble of DMD forecasts or the DMD of an ensemble of forecasts. We also discuss results from projecting / compressing high-fidelity forecasts using schemes such as POD projection and K-SVD for sparse representation (Section 2.5) due to showing promise for distributing forecasts to remote vehicles.

In this thesis we also compare several of these DMD methods and examine adaptive Bayesian data assimilation methods such as the Gaussian Mixture Model - Dynamically Orthogonal (GMM-DO) filter [109] combined with DMD to improve forecast accuracy based on observational data (taken either by the platform itself or a combination of data taken by the platform and other nearby vehicles/platforms that are communicated to the vehicle). The GMM-DO filter uses the stochastic Dynamically Orthogonal (DO) field equations and their adaptive stochastic subspace [30] to predict prior probabilities for the full dynamical state, effectively approximating the Fokker-Planck equation. At assimilation times, the DO subspace realizations are fit to semiparametric Gaussian Mixture Models (GMMs) using the Expectation-Maximization algorithm and the Bayesian Information Criterion. Bayes's law is then efficiently carried out analytically within the evolving stochastic subspace. We extend this approach to stochastic DMD forecasts and Bayesian GMM-DO updates of the DMD state and parameters, learning from the limited gappy observation data sets. In a novel fashion we combine DMD modeling methods with Ensemble Kalman Filter (EnKF) and the GMM-DO filter to produce DMD forecasts with Bayesian data assimilation that can be computed quickly and efficiently onboard an autonomous platform. We compare the accuracy of our GMM-DO results to traditional DMD forecasts and DMD with EnKF fore-



cast results and show that in Root Mean Square Error (RMSE) sense as well as error field sense, that the errors produced by DMD with GMM-DO technique are smaller and the errors grow slower in time than the other forecast methods mentioned. We demonstrate the error results of the DMD with GMM-DO technique utilizing sea surface temperature simulations of a 300 member ensemble set from the POSYDON POINT experiment. In virtually every instance, our forecasts using DMD with GMM-DO outperform DMD and DMD with EnKF in a RMSE, pattern correlation coefficient (PCC), and error field sense. In our results we showcase the DMD of the ensemble method due to its accurate and computationally efficient results, that we conclude, could be readily applied onboard autonomous platforms.

### 1.3 Marine Autonomy

Although not specifically addressed by the work in this thesis, path planning and marine autonomy have important ties to work we conduct.

The following is a summary of marine autonomy that is derived from [59]. Fixed platforms or expeditions with vessels have long been used to collect a variety of observations and data of the ocean environment [111][22][105]. Since the turn of the millennium, research expeditions have been supplemented and augmented with varied robotics-based platforms and sensors [100][5][91]. Some types of platforms used include vehicles with a relative speed, such as propelled autonomous underwater vehicles (AUVs), underwater gliders, wave gliders, solar-powered vehicles, and surface craft. Some experiments have also included vehicles with no or limited relative speeds such as autonomous drifters, floats, hybrid profilers, and semi-drifting surface craft. A crucial advantage of these robotic platforms is that they reduce the under-sampling of ocean surveys and augment other data sources (like satellite observations). One significant disadvantage of these platforms is they are impacted by ocean motions, such as currents and waves. The ocean observations taken by them are a mixture of Eulerian (fixed or not affected by currents) and Lagrangian (current-following) observations. Researchers have solidified and noted this mixture in the name “autonomous and Lagrangian platforms and sensors (ALPS),” (e.g., [100]; [23], and references therein). To account for the ocean motions and other effects, ALPS systems should become “expert systems” (e.g., [43]) that plan and optimize their motions, using and integrating predictive models, control algorithms, uncertainty estimates, and data assimilation [59]. Such integration of disciplines was

initiated by the Autonomous Ocean Sampling Network (e.g., [19]; [99]; [54]; [58]; [96]; [18]), and this area remains an area of active research.

One specific area of research where our reduced order model and data assimilation techniques could be used is in the area of optimal path planning. Conversely, the DMD and DA schemes here could benefit from utilizing various path planning methods. MSEAS has worked extensively on ocean path planning and continues to do so. In general, path planning techniques start from the governing path planning equations [77][76][71][60]. These equations account for the effects of the dynamic environmental ocean flow field on the net vehicle motion. The equations predict globally optimal paths without heuristics [75], are computationally efficient [77], and can account for forbidden or unsafe areas and stationary or moving obstacles [72]. For any given ocean currents (however strong), the level-set path planning equations provide the exact time-optimal paths to travel from one location to another. The methodology has been extended to energy-optimal paths [113][112], swarm-optimal paths [72], and paths that maximize the quality of observational data collected by the platform [70][61]. The algorithms accuracy and effectiveness were experimentally demonstrated in several sea-trials [112][25][86]. Time optimal path planning has been shown to be effective when planning multi-waypoint missions in dynamic ocean environments [31] and can be combined with decision theory to create a risk-optimal path [115]. Recently, MSEAS path planning algorithms have even been extended to estimate optimal harvesting of offshore macroalgae farming [7] and the path planning approach was extended to address both fully 3D realistic ocean setups and 3D propulsion constraints [50].

Effective predictive models of the relevant ocean dynamics and efficient uncertainty prediction and data assimilation schemes are crucial for quantitative optimal path planning and sampling. Additionally these forecast models and data assimilation schemes must be computationally inexpensive to allow for other tasks, like path planning to occur simultaneously or in rapid succession. This is where the need for computationally cheap and low storage forecast and data assimilation techniques, some of which we detail in this thesis, will play a critical role in the future for computing onboard autonomous platforms at sea.

## 1.4 Thesis Overview

This thesis utilizes several techniques for use with and onboard autonomous at sea platforms. Among these techniques are compression techniques, a burgeoning field known as Dynamic Mode Decomposition (DMD), which connects nonlinear dynamical systems with well-established methods in dynamical systems theory, and ensemble data assimilation methods. We utilize DMD to extract underlying dynamics from deterministic and/or stochastic inputs from simulated ocean environments. We combine DMD prediction model methods with ensemble based data assimilation (DA) techniques like Ensemble Kalman filter (EnKF) and Gaussian mixture models using the Dynamically Orthogonal field equations (GMM-DO).

Chapter 2 begins with a description of DMD and DMD methods that we have used for ocean environments as well as a discussion of data compression techniques. Due to the often large size of the state space for ocean environments, the extraction of DMD modes from high fidelity forecast simulations is best done off the vehicle (like on a reach-back cell). The DMD and Proper Orthogonal Decomposition (POD) modes and other preliminary data can then be pre-loaded on the platform. We focus on a handful of methods of DMD that could be useful depending on the desired application of the end user. We also discuss methods of compression that could be used to inexpensively compress snapshots or data that could be sent to a vehicle when communication bandwidth is limited. These compressed snapshots or data could be readily and inexpensively reconstructed on the platform as needed or used in subspace methods (without needing to be reconstructed) like data assimilation.

In Chapter 3 we describe ensemble based DA techniques primarily focusing on GMM-DO with some discussion of EnKF as well. Chapter 3 also discusses the novel approach in which we tie together DA (i.e. GMM-DO) with DMD for a data driven technique that can inexpensively be used onboard autonomous platforms.

Chapter 4 shows the results of combining DMD and DA techniques for a vorticity simulation of flow past an island/cylinder. Additionally we show results of using our technique on a high fidelity ensemble ocean simulation in the Middle Atlantic-New York Bight region that was generated for the POSYDON-POINT experiment by the Model Multidisciplinary Simulation, Estimation, and Assimilation Systems (MSEAS) research group at the Massachusetts Institute of Technology. In this example, we evaluate the techniques on sea surface tem-

perature. Currently, we evaluate variables individually. In future work, we plan to include multivariate solutions (after normalization) with our work, e.g., [55, 56]. Additionally, how to utilize multi-scale applications are being considered.

Chapter 5 discusses this thesis’s conclusions and highlights some areas for future work.

## Chapter 2

# Reduced Order Models and Dynamic Mode Decomposition

### 2.1 DMD Origins

In the 1930's Bernard Koopman [48] and Koopman and von Neumann [49] developed methods for interpreting and extracting underlying nonlinear dynamics from sets of inputs. Due to a lack of computational resources, the work of Koopman went largely unused for many decades. In the decade staring around the turn of the new millennium, interest in the Koopman theory was renewed by Mezić et al. [85] [84]. In the following, we describe some developments and methods that have come about since that time [41].

### 2.2 DMD Architecture

The DMD architecture normally considers a continuous-time dynamical system

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t; \mu) \quad (2.1)$$

where  $\mathbf{x}(t) \in \mathbb{R}^n$  is the vector representing the state of the system at time  $t$ ,  $\mu$  contains the parameters of the system, and  $\mathbf{f}(\cdot)$  is the dynamics. We are typically concerned with systems with  $n \gg 1$  ( $n$  representing the size of the state space), as such large systems correspond to the discretization of PDEs at many discrete locations in space. We can further discretize (2.1) in time or sample the solution at every  $\Delta t$ . Denoting the discrete time index by

subscript  $k$  such that  $\mathbf{x}_k = \mathbf{x}((k-1)\Delta t)$ , measurements (or estimates) of the system are collected at the discrete intervals from  $k = 1, 2, \dots, m$ . Numerical solutions are typically used to predict future states solutions of (2.1) as analytical solutions cannot often be determined. DMD uses an equation-free idea where the right hand side of (2.1) does not need to be known. Instead, past discrete solutions or direct measurements are used as inputs by DMD to approximate the dynamics and allow for future state prediction.

DMD constructs the approximate linear representation of (2.1) as

$$\frac{d\mathbf{x}}{dt} = \mathcal{A}\mathbf{x}. \quad (2.2)$$

If the initial condition is  $\mathbf{x}(0) = \mathbf{x}_1$ , then the solution is [10]

$$\mathbf{x}(t) = \sum_{i=1}^n \phi_i \exp(\omega_i t) b_i = \mathbf{\Phi} \exp(\mathbf{\Omega} \mathbf{t}) \mathbf{b}, \quad (2.3)$$

where  $\phi_i$  and  $\omega_i$  are the eigenvectors and eigenvalues of the matrix  $\mathcal{A}$  and  $b_k$  contains the coordinates of  $\mathbf{x}_1$  in the eigenvector basis. Methods for computing  $\mathbf{b}$  are discussed in descriptions of DMD methods in the following section.  $\mathbf{\Phi}$  is a matrix whose columns are made up by the eigenvectors  $\phi_i$  and  $\mathbf{\Omega}$  is the matrix whose diagonals are  $\omega_i$ .

From the given continuous dynamics (2.2), it is possible to construct a discrete-time system given by,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \quad (2.4)$$

where

$$\mathbf{A} = \exp(\mathcal{A}\Delta t). \quad (2.5)$$

Here,  $\mathcal{A}$  is the continuous-time dynamics matrix in (2.2) and  $\Delta t$  is the fixed interval between time steps. The eigenvectors and eigenvalues of  $\mathbf{A}$  are referred to as the DMD modes ( $\phi_j$ ) and DMD eigenvalues ( $\lambda_j$ ) respectively. The solution to (2.4) can then be given by

$$\mathbf{x}_{k+1} = \sum_{j=1}^n \phi_j \lambda_j^k b_j = \mathbf{\Phi} \mathbf{\Lambda}^k \mathbf{b}, \quad (2.6)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix with diagonal entries of  $\lambda_j$ .

DMD methods estimate the low-rank eigendecomposition of matrix  $\mathbf{A}$  such that

$$\|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k\|_2 \quad (2.7)$$

is minimized for times  $k = 1, 2, \dots, m-1$ . The optimality holds over the training window in which  $\mathbf{A}$  is constructed and can be used for future predictions beyond the window [41].

To minimize the error (2.7) using the sample set of snapshots from  $k = 1, 2, \dots, m$ , two matrices are formed (they will be the inputs to the DMD algorithms). For a sequential set of column vectors  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  where each  $\mathbf{x}_k \in \mathbb{R}^n$ , matrices  $\mathbf{X}$  and  $\mathbf{X}'$  are formed as follows

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix} \quad (2.8)$$

and

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \quad (2.9)$$

Considering  $\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k)$  where  $\mathbf{F}$  is the map corresponding to the evolution of (2.1), DMD computes the eigenvalues and eigenvectors of the best-fit linear operator  $\mathbf{A}$  that relates  $\mathbf{X}' \approx \mathbf{A}\mathbf{X}$ . If the size of the state space is small,  $\mathbf{A}$  could be computed as  $\mathbf{A} = \mathbf{X}'\mathbf{X}^\dagger$ , where  $^\dagger$  indicates the Moore-Penrose pseudoinverse. This is not normally practical when the state space is large, so DMD methods provide alternatives to find the eigendecomposition of  $\mathbf{A}$  which then allows for future-state predictions.

## 2.3 DMD Methods

In this thesis, we utilize and compare several methods for reduced-order regional ocean prediction. Although most of the DMD methods can be used with imaginary inputs, we denote the algorithms assuming real inputs since the ocean simulation inputs and idealized inputs we use consist of only real parts. Much of this summary is taken from [41] which uses a common notation for the DMD methods.

### 2.3.1 Projected DMD

For projected DMD the snapshots must be in order (which is not a requirement for some other methods such as exact DMD). The algorithm produces a low-rank matrix projected onto Proper Orthogonal Decomposition (POD) modes to improve efficiency when computing the eigenvectors (DMD modes) and eigenvalues of the time-stepping matrix  $\mathbf{A}$  [51]. The DMD modes are projected onto POD modes, hence the name projected DMD.

#### Algorithm: Projected DMD

1. Arrange the inputs into sequential snapshot matrices  $\mathbf{X}$  and  $\mathbf{X}'$  as in (2.8) and (2.9).
2. Compute the compact Singular Value Decomposition (SVD) of  $\mathbf{X}$  such that

$$\mathbf{X} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T \quad (2.10)$$

where  $\mathbf{U}_r \in \mathbb{R}^{n \times r}$ ,  $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$  and,  $\mathbf{V}_r \in \mathbb{R}^{(m-1) \times r}$ ,  $r$  is the reduced rank of  $\mathbf{X}$ , and  $\mathbf{V}_r^T$  denotes the transpose of matrix  $\mathbf{V}_r$ . The matrix  $\mathbf{U}_r$  (the left singular vectors) are the proper orthogonal decomposition (POD) modes.

3. The matrix  $\mathbf{A}$  could be computed as follows

$$\mathbf{A} = \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \mathbf{U}_r^T. \quad (2.11)$$

In practice though, computing  $\mathbf{A}$  is extremely expensive so instead, it is much more computationally efficient to define a matrix  $\tilde{\mathbf{A}}$  which is the  $r \times r$  projection of  $\mathbf{A}$  onto POD modes as

$$\tilde{\mathbf{A}} \equiv \mathbf{U}_r^T \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1}. \quad (2.12)$$

where the left/right multiply of (2.11) by  $\mathbf{U}_r^T / \mathbf{U}_r$  projected onto the POD modes.

4. The eigenvalues and eigenvectors of  $\tilde{\mathbf{A}}$  are determined by

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \mathbf{\Lambda}, \quad (2.13)$$

where column vectors of  $\mathbf{W}$  are the eigenvectors of  $\tilde{\mathbf{A}}$  and diagonals of  $\mathbf{\Lambda}$  are the corresponding eigenvalues  $\lambda_j$ . The eigenvalues of  $\tilde{\mathbf{A}}$  are also the non-zero eigenvalues of the much larger matrix  $\mathbf{A}$ . The projected DMD modes (which are the estimated non-zero eigenvectors



of the full matrix  $\mathbf{A}$ ) are given by the column vectors of the matrix,

$$\Phi = \mathbf{U}_r \mathbf{W}. \quad (2.14)$$

5. To perform state reconstruction or future-state prediction, for convenience we define a matrix  $\Omega$  whose diagonal entries are made up by  $\omega_j = \frac{\ln(\lambda_j)}{\Delta t}$ . Then the predicted DMD solution at some time  $t$  is given by

$$\mathbf{x}(t) \approx \Phi \exp(\Omega t) \mathbf{b}. \quad (2.15)$$

Here  $\mathbf{b}$  is a vector of DMD amplitudes (which are sometimes referred to as DMD coefficients). If the initial snapshot  $\mathbf{x}_1$  is at time  $t_1 = 0$ , then  $\mathbf{x}_1 = \Phi \mathbf{b}$  and thus

$$\mathbf{b} = \Phi^\dagger \mathbf{x}_1. \quad (2.16)$$

This can be expensive if the size of  $\mathbf{x}_1$  is large. As an alternative, DMD amplitudes can be calculated much more inexpensively using POD projected data [13]. If we consider  $\tilde{\mathbf{A}}$  defines the linear model for the dynamics such that  $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}} \tilde{\mathbf{x}}_k$ , then we can compute the DMD amplitudes using the following:

$$\mathbf{x}_1 = \Phi \mathbf{b} \quad (2.17a)$$

$$\mathbf{U}_r \tilde{\mathbf{x}}_1 = \mathbf{X}' \mathbf{V}_r \Sigma_r^{-1} \mathbf{W} \mathbf{b} \quad (2.17b)$$

$$\tilde{\mathbf{x}}_1 = \mathbf{U}_r^T \mathbf{X}' \mathbf{V}_r \Sigma_r^{-1} \mathbf{W} \mathbf{b} \quad (2.17c)$$

$$\tilde{\mathbf{x}}_1 = \tilde{\mathbf{A}} \mathbf{W} \mathbf{b} \quad (2.17d)$$

$$\tilde{\mathbf{x}}_1 = \mathbf{W} \Lambda \mathbf{b} \quad (2.17e)$$

$$\mathbf{b} = (\mathbf{W} \Lambda)^{-1} \tilde{\mathbf{x}}_1. \quad (2.17f)$$

### 2.3.2 Exact DMD

Exact DMD solves the same problem as projected DMD but computes the exact DMD modes of  $\mathbf{A}$  rather than projecting onto POD modes [51]. These exact modes are determined all while still avoiding explicitly computing  $\mathbf{A}$ . Indeed, as showed by Tu et. al. [119], the dominant  $r$  exact eigenvectors of matrix  $\mathbf{A}$  can be computed from the eigendecomposition of

$\tilde{\mathbf{A}}$ . Besides determining the exact DMD modes, as long as input pairs  $\mathbf{x}_k$  and  $\mathbf{x}_{k+1}$  are in the same columns of  $\mathbf{X}$  and  $\mathbf{X}'$  respectively, another benefit is that the calculated eigenvalues and eigenvectors of  $\mathbf{A}$  will be the same regardless of the order of the snapshot pairs.

**Algorithm: Exact DMD**

1. Arrange the inputs into snapshots matrices  $\mathbf{X}$  and  $\mathbf{X}'$  as in (2.8) and (2.9).
2. Perform SVD as in equation (2.10), compute  $\tilde{\mathbf{A}}$  as in (2.12), and perform eigendecomposition as in (2.13).
3. The exact DMD modes are given by the column vectors of the matrix

$$\Phi = \mathbf{X}' \mathbf{V}_r \Sigma_r^{-1} \mathbf{W}. \quad (2.18)$$

4. The state solution at time  $t$  can be predicted using (2.15).

### 2.3.3 Compressed DMD

Compressed DMD solves the Euclidean norm minimization problem, but it compresses the inputs first [14]. With matrices  $\mathbf{X}$  and  $\mathbf{X}'$ , it is possible to compress the inputs, compute DMD on the compressed inputs and reconstruct DMD modes and eigenvalues of the full-state by linearly combining full-state snapshots according to the compressed DMD transformation. This DMD computation is much faster than DMD on full-state data particularly for large data sets. It should be noted that the compressed DMD here refers to the spatial compression.

Compressed DMD relies on two essential conditions. First, the snapshots have to be sparse in some basis given by the columns of  $\mathbf{\Gamma}$ , so that  $\mathbf{X} = \mathbf{\Gamma} \mathbf{S}$  and  $\mathbf{X}' = \mathbf{\Gamma} \mathbf{S}'$ . Here,  $\mathbf{S}$  and  $\mathbf{S}'$  have sparse columns and are considered the sparse portions of  $\mathbf{X}$  and  $\mathbf{X}'$  respectively. The basis  $\mathbf{\Gamma} \in \mathbb{R}^{n \times n}$  can be Fourier, wavelet, or it can be the first  $p$ -dominant POD modes found by the initial SVD (a potential implementation strategy would be to re-evaluate the POD modes from time to time to reflect the change in the dynamics) [11]. Next, we consider a (pseudo)-measurement matrix  $\mathbf{C} \in \mathbb{R}^{p \times n}$ , where  $p < n$ , that must be incoherent with respect to the sparse basis  $\mathbf{\Gamma}$ , i.e. rows of  $\mathbf{C}$  are uncorrelated with columns of  $\mathbf{\Gamma}$ . This will hold true generally as long as  $\mathbf{C}$  is a Gaussian random measurement matrix. If we assume

each column of  $\mathbf{X}$  and  $\mathbf{X}'$  is in the same sparse subspace of the basis  $\mathbf{\Gamma}$ , then we can ensure that the POD modes and DMD modes will also be in the same sparse subspace.

**Algorithm: Compressed DMD**

1. Arrange the inputs into snapshots matrices  $\mathbf{X}$  and  $\mathbf{X}'$  as in (2.8) and (2.9).
2. Determine the reduced number of (pseudo)-measurements to be used,  $p$ , and use a random matrix (or Gaussian random matrix)  $\mathbf{C} \in \mathbb{R}^{p \times n}$  to compress the inputs and obtain reduced sets as follows:

$$\mathbf{Y} = \mathbf{C}\mathbf{X} \text{ and } \mathbf{Y}' = \mathbf{C}\mathbf{X}'. \quad (2.19)$$

3. Compute exact DMD or projected DMD on  $(\mathbf{Y}, \mathbf{Y}')$  as in (2.10), (2.12), (2.13), and (2.18) replacing  $\mathbf{X}$  with  $\mathbf{Y}$  and  $\mathbf{X}'$  with  $\mathbf{Y}'$  to obtain  $(\mathbf{\Lambda}_{\mathbf{Y}}, \mathbf{W}_{\mathbf{Y}})$  and  $\mathbf{\Phi}_{\mathbf{Y}}$ .
4. The full-state DMD modes can then be constructed using

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V}_{\mathbf{Y}}\mathbf{\Sigma}_{\mathbf{Y}}^{-1}\mathbf{W}_{\mathbf{Y}} \quad (2.20)$$

5. With the DMD modes  $(\mathbf{\Phi})$  and the DMD eigenvalues (diagonal entries of  $\mathbf{\Lambda}_{\mathbf{Y}}$ ), the DMD future-state prediction for the full-state uses (2.15).

### 2.3.4 Total DMD

Conventional DMD methods often fail to accurately capture the underlying dynamics when snapshot data contain significant sensor noise. Total DMD projects each input time snapshot and its sequential pair onto a joint subspace. When the sensor noise remains mostly in the orthogonal complement of the joint subspace, this reduces noise in the inputs to the DMD algorithm. This leads to a slight cost increase but allows DMD to capture the dynamical descriptions much more effectively. Conventional DMD methods (such as exact and projected) minimize errors with respect to the time-shifted matrix  $\mathbf{X}'$  only. This yields a biased analysis when the snapshots exhibit noise. Total DMD, however, minimizes the orthogonal distance between the linear fit involving both initial  $\mathbf{X}$  and final  $\mathbf{X}'$  states, allowing a de-biased analysis [39].

Mathematically, instead of minimizing the residual  $\|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$  as in other methods,

Total DMD [39] minimizes  $\left\| \begin{bmatrix} \Delta \mathbf{X} \\ \Delta \mathbf{X}' \end{bmatrix} \right\|_F$  subject to  $\mathbf{X}' + \Delta \mathbf{X}' = \mathbf{A}(\mathbf{X} + \Delta \mathbf{X})$ . This total-least-squares problem can then be solved by projecting  $\mathbf{X}$  and  $\mathbf{X}'$  onto an augmented subspace created from the matrix  $\begin{bmatrix} \mathbf{X} \\ \mathbf{X}' \end{bmatrix}$ .

**Algorithm: Total DMD**

1. Arrange the inputs into snapshots matrices  $\mathbf{X}$  and  $\mathbf{X}'$  as in (2.8) and (2.9).
2. Determine the best subspace by compact SVD of an augmented matrix  $\mathbf{Z}$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}' \end{bmatrix} = \mathbf{U}_Z \mathbf{\Sigma}_Z \mathbf{V}_Z^T. \quad (2.21)$$

3. Project  $\mathbf{X}$  and  $\mathbf{X}'$  onto the subspace, resulting in de-biased / reduced inputs  $\mathbf{Y}$  and  $\mathbf{Y}'$  as follows:

$$\mathbf{Y} = \mathbf{X} \mathbf{V}_Z \mathbf{V}_Z^T \text{ and } \mathbf{Y}' = \mathbf{X}' \mathbf{V}_Z \mathbf{V}_Z^T. \quad (2.22)$$

4. Perform DMD using (2.10), (2.12), (2.13), and (2.18) except that wherever  $\mathbf{X}$  is used substitute  $\mathbf{Y}$  and wherever  $\mathbf{X}'$  is used substitute  $\mathbf{Y}'$ . Instead of (2.18), (2.14) could be used if projected DMD modes are preferred over the exact DMD modes.
5. Use (2.15) to compute the state space prediction at time  $t$ .

### 2.3.5 Optimized DMD

A pitfall of many of the DMD methods (e.g. projected, exact, compressed) is that the computed eigenvalues are biased by the presence of sensor noise in the inputs (only residuals of the linear system are minimized). Total DMD is one of the de-biasing methods that tries to overcome this by minimizing both the initial and final errors. Alternatively, Optimized DMD [3] addresses the issues of biases due to sensor noise by creating an optimization problem where the identified linear operator has a fixed rank.

The standard DMD methods treat the data pairwise, snapshot to snapshot, rather than as a whole, and favor one direction (forward in time). Optimized DMD allows the reconstruction errors to be distributed throughout, minimizing noise.

### Algorithm: Optimized DMD

1. Arrange inputs into a single matrix made up of all the snapshots:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \quad (2.23)$$

2. Suppose  $\mathbf{x}(t)$  is the solution to  $\dot{\mathbf{x}}(t) = \mathcal{A}\mathbf{x}(t)$ . With initial condition  $\mathbf{x}_1$ , the analytical solution is:

$$\mathbf{x}(t) = \exp(\mathcal{A}t)\mathbf{x}_1. \quad (2.24)$$

3. Assume that the matrix  $\mathcal{A}$  is diagonalizable such that  $\mathcal{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$  where  $\mathbf{S} \in \mathbb{R}^{n \times r}$  and  $\mathbf{\Lambda} \in \mathbb{R}^{r \times r}$  for a target rank  $r$ . Therefore,  $\mathbf{x}(t)$  can be rewritten:

$$\mathbf{x}(t) \approx \mathbf{S} \exp(\mathbf{\Lambda}t) \mathbf{S}^\dagger \mathbf{x}_1. \quad (2.25)$$

4. If we let the diagonals of  $\mathbf{\Lambda}$  be given by  $\alpha_1, \alpha_2, \dots, \alpha_r$  and define the matrix basis function or time matrix  $\mathbf{\Omega}(\alpha)$  with entries  $\Omega(\alpha)_{k,j} = \exp(\alpha_j t_k)$  for  $m$  sample times  $t_k$ ,  $\mathbf{X}^T$  can be written:

$$\mathbf{X}^T \approx \mathbf{\Omega}(\alpha) \mathbf{B}, \quad (2.26)$$

where  $B_{i,j} = S_{j,i} (\mathbf{S}^\dagger \mathbf{x}_1)_i$  are the entries of  $\mathbf{B}$ . Here,  $S_{j,i}$  is the  $j$ -th row and  $i$ -th column of matrix  $\mathbf{S}$  and  $(\mathbf{S}^\dagger \mathbf{x}_1)_i$  is the  $i$ -th entry of the vector  $(\mathbf{S}^\dagger \mathbf{x}_1)$ .

5. Suppose  $\hat{\alpha}$  and  $\hat{\mathbf{B}}$  are solutions to minimization of  $\|\mathbf{X}^T - \mathbf{\Omega}(\alpha) \mathbf{B}\|_F$  (solved using a variable projection algorithm [36, 46]).

6. Optimized DMD eigenvalues are defined by  $\lambda_i = \hat{\alpha}_i$  and the DMD modes are

$$\phi_i = \frac{1}{\|\hat{\mathbf{B}}^T(:, i)\|_2} \hat{\mathbf{B}}^T(:, i), \quad (2.27)$$

where  $\hat{\mathbf{B}}^T(:, i)$  is the  $i$ -th column of  $\hat{\mathbf{B}}^T$ .

7. If we define  $b_i = \|\hat{\mathbf{B}}^T(:, i)\|_2$ , future-state prediction at time  $t_k$  can then be made with

the following equation:

$$\mathbf{x}_k = \sum_{i=1}^r b_i \exp(\lambda_i t_k) \phi_i. \quad (2.28)$$

### 2.3.6 Streaming DMD

Most DMD methods view DMD as a post-processing tool, meaning we require a large number of inputs to extract spatial and temporal modes for analysis. There are many instances where an online and incrementally updated method would be advantageous. A streaming method, where a snapshot pair is evaluated as it is received, can minimize data storage [40]. Streaming DMD allows for DMD to be performed incrementally as new snapshot pairs are available. We arrange snapshot pairs  $\mathbf{x}_k \in \mathbb{R}^n$  and  $\mathbf{x}_{k+1} \in \mathbb{R}^n$  that are spaced a fixed time interval apart ( $\Delta t$ ) stored in  $\mathbf{X}$  and  $\mathbf{X}'$  as in (2.8) and (2.9).

First we must compute a matrix  $\mathbf{Q}_{\mathbf{x}} \in \mathbb{R}^{n \times r}$  (where  $r$  is the reduced rank of  $\mathbf{X}$  and  $\mathbf{X}'$ ) whose columns form orthonormal basis for the image of  $\mathbf{X}$ . The DMD operator is then

$$\mathbf{K} = \mathbf{Q}_{\mathbf{x}} \tilde{\mathbf{K}} \mathbf{Q}_{\mathbf{x}}^T, \quad (2.29)$$

where  $\tilde{\mathbf{K}}$  is a  $r \times r$  matrix defined by

$$\tilde{\mathbf{K}} \equiv \mathbf{Q}_{\mathbf{x}}^T \mathbf{X}' \mathbf{X}^\dagger \mathbf{Q}_{\mathbf{x}}. \quad (2.30)$$

The DMD modes and eigenvalues are the eigenvectors and eigenvalues of  $\mathbf{K}$  which may be computed from the much smaller matrix  $\tilde{\mathbf{K}}$ .

For streaming DMD situations we assume that we only have access to a pair of snapshots  $(\mathbf{x}_k, \mathbf{x}_{k+1})$  at time  $k$ . There is then an alternate way to compute  $\tilde{\mathbf{K}}$  to allow for it to be incrementally updated as new snapshots are available. To do this, we need to determine the orthonormal basis for  $\mathbf{X}$  and  $\mathbf{X}'$  creating column matrices  $\mathbf{Q}_{\mathbf{x}} \in \mathbb{R}^{n \times r}$  and  $\mathbf{Q}_{\mathbf{x}'} \in \mathbb{R}^{n \times r}$ . Next, we project  $\mathbf{X}$  and  $\mathbf{X}'$  onto the respective orthogonal base  $\mathbf{Y} \equiv \mathbf{Q}_{\mathbf{x}}^T \mathbf{X}$  and  $\mathbf{Y}' \equiv \mathbf{Q}_{\mathbf{x}'}^T \mathbf{X}'$ . Then, we can define new matrices  $\mathbf{C} = \mathbf{Y}' \mathbf{Y}^T$ ,  $\mathbf{G}_{\mathbf{x}} = \mathbf{Y} \mathbf{Y}^T$ , and  $\mathbf{G}_{\mathbf{x}'} = \mathbf{Y}' \mathbf{Y}'^T$  which allows us to rewrite  $\tilde{\mathbf{K}}$  as

$$\tilde{\mathbf{K}} = \mathbf{Q}_{\mathbf{x}}^T \mathbf{Q}_{\mathbf{x}'} \mathbf{C} \mathbf{G}_{\mathbf{x}}^\dagger. \quad (2.31)$$

Advantages of this formulation is that  $\tilde{\mathbf{K}}$  can be updated incrementally and much less storage is required particularly for streams where  $m$  is already extremely large and increasing. Methods that learn subspaces and complete rank updates can be used [12, 57, 104, 95, 94, 29].

### Algorithm: Streaming DMD

As a new snapshot pair is acquired, an iteration to update the DMD is performed as follows:

1. For the new snapshot pair  $\mathbf{x}_k$  and  $\mathbf{x}_{k+1}$ , compute the residuals

$$e_{\mathbf{X}} = (\mathbf{I} - \mathbf{Q}_{\mathbf{X}} \mathbf{Q}_{\mathbf{X}}^T) \mathbf{x}_k \quad (2.32)$$

and

$$e_{\mathbf{X}'} = (\mathbf{I} - \mathbf{Q}_{\mathbf{X}'} \mathbf{Q}_{\mathbf{X}'}^T) \mathbf{x}_{k+1} \quad (2.33)$$

where  $\mathbf{I}$  is the appropriate dimension Identity matrix.

2. If  $\|e_{\mathbf{X}}\| > \epsilon$  or  $\|e_{\mathbf{X}'}\| > \epsilon$  (where  $\epsilon$  is a user defined tolerance), the dimension of the corresponding basis,  $\mathbf{Q}_{\mathbf{X}}$  or  $\mathbf{Q}_{\mathbf{X}'}$ , is increased by appending an additional column  $e_{\mathbf{X}}/\|e_{\mathbf{X}}\|$  or  $e_{\mathbf{X}'}/\|e_{\mathbf{X}'}\|$  respectively.  $\mathbf{G}_{\mathbf{X}}$ ,  $\mathbf{G}_{\mathbf{X}'}$ , and  $\mathbf{C}$  must be zero-padded to maintain dimensional consistency.

3. If either basis,  $\mathbf{Q}_{\mathbf{X}}$  or  $\mathbf{Q}_{\mathbf{X}'}$ , becomes too large ( $r > r_0$ ), compute leading eigenvectors of  $\mathbf{G}_{\mathbf{X}}$  and  $\mathbf{G}_{\mathbf{X}'}$  ( $\mathbf{W}_{\mathbf{X}}$  and  $\mathbf{W}_{\mathbf{X}'}$  respectively), then set

$$\mathbf{G}_{\mathbf{X}} \leftarrow \mathbf{W}_{\mathbf{X}}^T \mathbf{G}_{\mathbf{X}} \mathbf{W}_{\mathbf{X}},$$

$$\mathbf{G}_{\mathbf{X}'} \leftarrow \mathbf{W}_{\mathbf{X}'}^T \mathbf{G}_{\mathbf{X}'} \mathbf{W}_{\mathbf{X}'},$$

$$\mathbf{C} \leftarrow \mathbf{W}_{\mathbf{X}'}^T \mathbf{C} \mathbf{W}_{\mathbf{X}},$$

$$\mathbf{Q}_{\mathbf{X}} \leftarrow \mathbf{Q}_{\mathbf{X}} \mathbf{W}_{\mathbf{X}},$$

and

$$\mathbf{Q}_{\mathbf{X}'} \leftarrow \mathbf{Q}_{\mathbf{X}'} \mathbf{W}_{\mathbf{X}'}.$$

Here,  $r_0$  is a pre-specified maximum allowable matrix rank at which the truncation step occurs.

4. Next, set

$$\mathbf{y}_k = \mathbf{Q}_{\mathbf{X}}^T \mathbf{x}_k \quad (2.34)$$

and

$$\mathbf{y}_{k+1} = \mathbf{Q}_{\mathbf{X}'}^T \mathbf{x}_{k+1}. \quad (2.35)$$

Then, let

$$\mathbf{G}_{\mathbf{X}} \leftarrow \mathbf{G}_{\mathbf{X}} + \mathbf{y}_k \mathbf{y}_k^T,$$

$$\mathbf{G}_{\mathbf{X}'} \leftarrow \mathbf{G}_{\mathbf{X}'} + \mathbf{y}_{k+1} \mathbf{y}_{k+1}^T,$$

and

$$\mathbf{C} \leftarrow \mathbf{C} + \mathbf{y}_{k+1} \mathbf{y}_k^T.$$

5. If DMD modes and/or eigenvalues are needed at the end of an iteration, compute the eigenvectors and eigenvalues of  $\mathbf{C} \mathbf{G}_{\mathbf{X}}^\dagger$ . If  $\mathbf{v}_j$  is the  $j$ -th eigenvector of  $\mathbf{C} \mathbf{G}_{\mathbf{X}}^\dagger$  then  $\mathbf{Q}_{\mathbf{X}} \mathbf{v}_j$  is the  $j$ -th DMD mode.

6. Now with the DMD modes and eigenvalues, DMD future-state prediction or reconstruction can take place with equations (2.15) and (2.16).

## 2.4 DMD of the Ensemble

We utilized another approach to combine ensemble members to make probabilistic DMD predictions [119]. With a total of  $N$  ensemble members and  $m$  times, each snapshot can be represented as  $\mathbf{x}_k^q$  where  $k$  is the snapshot time and  $q$  is the  $q$ -th ensemble member. DMD can then be applied to matrices

$$\mathbf{X} = \begin{bmatrix} | & & | & | & & | & & | & | \\ \mathbf{x}_1^1 & \cdots & \mathbf{x}_{m-1}^1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_{m-1}^2 & \cdots & \mathbf{x}_1^N & \cdots & \mathbf{x}_{m-1}^N \\ | & & | & | & & | & & | & & | \end{bmatrix} \quad (2.36)$$



and

$$\mathbf{X}' = \begin{bmatrix} | & & | & | & & | & & | \\ \mathbf{x}_2^1 & \cdots & \mathbf{x}_m^1 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_m^2 & \cdots & \mathbf{x}_2^N & \cdots & \mathbf{x}_m^N \\ | & & | & | & & | & & | \end{bmatrix}. \quad (2.37)$$

This approach allows for extracting the DMD modes and eigenvalues from the entire ensemble set (or a specified number of ensemble members) simultaneously. With the overall ensemble DMD modes and eigenvalues, we can make prediction of a particular ensemble member using an initial condition (used to calculate the DMD amplitudes). We call this method DMD of the Ensemble. The results of [41] indicate that using multiple ensemble members can capture the underlying dynamics better than just using individual ensemble members. Having the modes extracted using a portion of the ensemble reduces storage and computation costs compared to using DMD modes of each ensemble individually. This could prove advantageous for unmanned vehicles especially when the number of realizations/ensemble members is extremely large. The DMD mode could be computed off the vehicle, then a reduced number of DMD modes could be sent to the vehicle for relatively inexpensive predictions with reduced communication and storage needs. In the following chapters, if the method of DMD used is not specified then the method implemented was DMD of the ensemble (using projected DMD modes). The main reason that projected DMD was used extensively was due to the ability to extend its linearity to data assimilation schemes, like GMM-DO, readily. The other methods of DMD were described above to show that there is a wide array of DMD methods that can be chosen based on user needs.

## 2.5 Reduced Order Projection and Subspace Methods

The above DMD methods are commonly utilized when high-fidelity model predictions are not available or too expensive to compute, and when enough past observations are available to directly build reduced order models. These reduced order models are then integrated forward in time to compute the forecasts. In specific marine applications, high-fidelity forecasts computed on land or a ship based facility are however often available. In this case, sending these forecasts efficiently to communication-limited autonomous platforms is one of the challenges. One approach to achieve this is to project the forecasts onto a suitable subspace and to only transmit the projected coefficients to the autonomous platforms. There are many options for such projections [52, 51], several of which based on singular value decomposition

(SVD), including adaptive SVD [4, 47, 21]. In the stochastic case, the adaptive SVD methods become related to DO decompositions [112, 116, 29]. We briefly review next some of the classic projection and subspace methods.

### 2.5.1 POD, PCA, and EOF Projections

Proper Orthogonal Decomposition (POD) is a method of analysis that identifies the dominant structures in a data-set [117]. In the above DMD notation, the dominant POD modes are the columns of  $\mathbf{U}_r$  that are determined from the SVD of the matrix  $\mathbf{X}$ , as in (2.10). These modes, or basis functions, capture as much energy of the system as possible [118]. POD is a commonly used model reduction technique, often based on snapshot data only. In the POD literature, Principal Component Analysis (PCA), and Empirical Orthogonal Functions [44] are often used to indicate similar basis functions, but POD modes are not necessarily mean subtracted as PCA models are (which is also referred to as EOF analysis when based on empirical data only [90]).

Due to the limited storage and power available on autonomous vehicles, projection methods such as POD can prove extremely useful for sending reduced forecasts to an autonomous vehicle. Sending the full-state data is indeed often impractical or impossible. To reduce the state space at time  $k$  using POD modes, one option is to project the state-space onto POD modes

$$\tilde{\mathbf{x}}_k = \mathbf{U}_r^T \mathbf{x}_k, \quad (2.38)$$

where  $\tilde{\mathbf{x}}_k \in \mathbb{R}^r$  and  $\mathbf{U}_r \in \mathbb{R}^{n \times r}$ . The state space can be reconstructed by  $\mathbf{x}_k = \mathbf{U}_r \tilde{\mathbf{x}}_k$ . If the POD modes from a set of past training data (columns of  $\mathbf{U}_r$ ) were pre-loaded onto a vehicle, this subspace could be used to reconstruct full-state data with future predictions that were inexpensively sent to the vehicle. This is advantageous because in the ocean the full-state dimensions are often  $\mathcal{O}(10^5 - 10^9)$  where as the reduced rank  $r$  is often  $\mathcal{O}(10^1 - 10^3)$ .

### 2.5.2 K-SVD and other sparse Projections

K-SVD allows for dictionary learning to create a dictionary for sparse representation of a signal [1] (note our notation differs from the K-SVD literature for consistency with the prior notation). There continues to be a growing interest in the study of sparse representation of signals based on an over-complete dictionary. Let  $\mathbf{x} \in \mathbb{R}^n$  be an observed signal, and the dic-

tionary  $\mathbf{D} \in \mathbb{R}^{n \times K}$ , and  $\mathbf{y} \in \mathbb{R}^K$  are the representation coefficients. We assume that without noise,  $\mathbf{x} = \mathbf{D}\mathbf{y}$ , where the vector  $\mathbf{y}$  is sparse. The goal of dictionary learning is to learn an over-complete dictionary,  $\mathbf{D}$ , that contains  $K$  signal-atoms. The sparse representation is the solution to

$$\min_y \|\mathbf{y}\|_0 \text{ subject to } \mathbf{x} = \mathbf{D}\mathbf{y} \quad (2.39)$$

or

$$\min_y \|\mathbf{y}\|_0 \text{ subject to } \|\mathbf{x} - \mathbf{D}\mathbf{y}\|_2 \leq \epsilon \quad (2.40)$$

where  $\epsilon$  is some small tolerance. To choose the best possible code-book,  $\mathbf{D}$ , we solve

$$\min_{\mathbf{D}, \mathbf{Y}} \|\mathbf{D}\mathbf{Y} - \mathbf{X}\|_F^2 \text{ subject to } \|\mathbf{y}_i\|_0 \leq T \quad (2.41)$$

where  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]$  is the collection of  $m$  observations,  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]$  is the collection of  $m$  representation coefficient vectors, and  $T$  is a constraint on the number of non-zero entries allowed in a column of  $\mathbf{D}$ . Now the problem can be solved by alternating minimization

$$\mathbf{Y}^{(j+1)} = \min_y \left\| \mathbf{D}^{(j)} \mathbf{Y} - \mathbf{X} \right\|_F^2 \text{ subject to } \|\mathbf{y}_i\|_0 \leq T. \quad (2.42)$$

and

$$\mathbf{D}^{(j+1)} = \min_D \left\| \mathbf{D} \mathbf{Y}^{(j+1)} - \mathbf{X} \right\|_F^2. \quad (2.43)$$

In words, at each iteration the sparse representation  $\mathbf{Y}$  and the dictionary  $\mathbf{D}$  are updated. The K-SVD algorithm may be used for compression of large ocean data [8, 33] and enable efficient transmission of ocean forecast and acoustic data to remote vehicles with bandwidth-limited, disadvantaged communications links.

## 2.6 DMD and Projection Results and Discussions

We now showcase the application of reduced order methods to two regional stochastic ocean forecasting experiments. Specifically, we apply several of the above DMD and SVD-based reduction methods, and discuss these results for use by simulated underwater vehicles in uncertain scenarios.

### 2.6.1 DMD Method Predictions

First, we consider a 12-hr forecast period (August 27, 2018 00Z) in the POSYDON-POINT experiment [65], and compare the performance of the SST forecasts of the above DMD methods with respect to the persistence forecast. All of the DMD forecasts were made from the first 85 hourly ocean simulation snapshots (00Z August 23 to 12Z August 26, 2018). The true forecast is assumed to be that of the full ocean modeling system [38].

The pattern correlation coefficients (PCCs) [55] for the Sea Surface Temperature (SST) forecasts of DMD methods are given in Table 2.1. From forecast hour 2 and beyond, all DMD methods beat persistence. The DMD algorithms were also compared for salinity and velocity, and at different times and depths, with similar results (not shown here). For its low cost, Compressed DMD performed well. Overall, Exact DMD performed best, followed by Compressed DMD, Optimized DMD, and Total DMD.

PCC							
Time (hr)	Persistence	Exact DMD	Total DMD	Projected DMD	Compressed DMD (90% Compressed)	Optimized DMD	Streaming DMD
0	1	0.99	1	1	0.99	1	1
1	0.98	0.99	0.99	0.99	0.99	1	0.99
2	0.93	0.98	0.97	0.97	0.99	0.98	0.97
3	0.85	0.97	0.94	0.94	0.97	0.95	0.94
4	0.77	0.94	0.91	0.91	0.94	0.92	0.91
5	0.69	0.91	0.88	0.88	0.9	0.89	0.88
6	0.63	0.88	0.86	0.86	0.87	0.87	0.86
7	0.58	0.85	0.83	0.83	0.83	0.84	0.83
8	0.55	0.82	0.8	0.8	0.81	0.82	0.8
9	0.53	0.8	0.77	0.77	0.78	0.8	0.77
10	0.53	0.78	0.75	0.75	0.88	0.79	0.75
11	0.54	0.76	0.74	0.74	0.85	0.76	0.74
12	0.55	0.75	0.72	0.72	0.74	0.73	0.72

Table 2.1: Pattern correlation coefficients (PCC) of SST errors vs. forecast time for the ensemble member 100 of the POSYDON-POINT experiment, for seven DMD methods. Green indicates PCC values above 0.8, blue PCC values between 0.6 and 0.8, and red PCC values below 0.6.

## 2.6.2 Splitting the Domain into Regions

To improve future-state prediction using DMD, we found that in most instances, splitting the domain into separate regions where relatively distinct or independent dynamics occurred was a good idea. In the following example, we separate the domain in two: the shelf region and the slope and deep water region. DMD methods were employed on SST from each domain on individual ensemble members to compute 12 hour forecasts using 85 hour training snapshots. The results are shown in Fig. 2-1. The 12 hour future SST predictions had PCC values of 0.54 for persistence, 0.61 for Total DMD on the entire domain simultaneously, and 0.75 for Total DMD on the shelf/deep water separately. The smaller domain for each of the regions means that the cost of DMD on the split domain and DMD on the entire domain were similar. We found similar results on other tests cases. In general, they imply that regional DMD forecasts may be improved by using a multi-domain approach. This would prove beneficial when dealing with autonomous vehicles operating in these specific regions, without capabilities for much inter-vehicle communication.

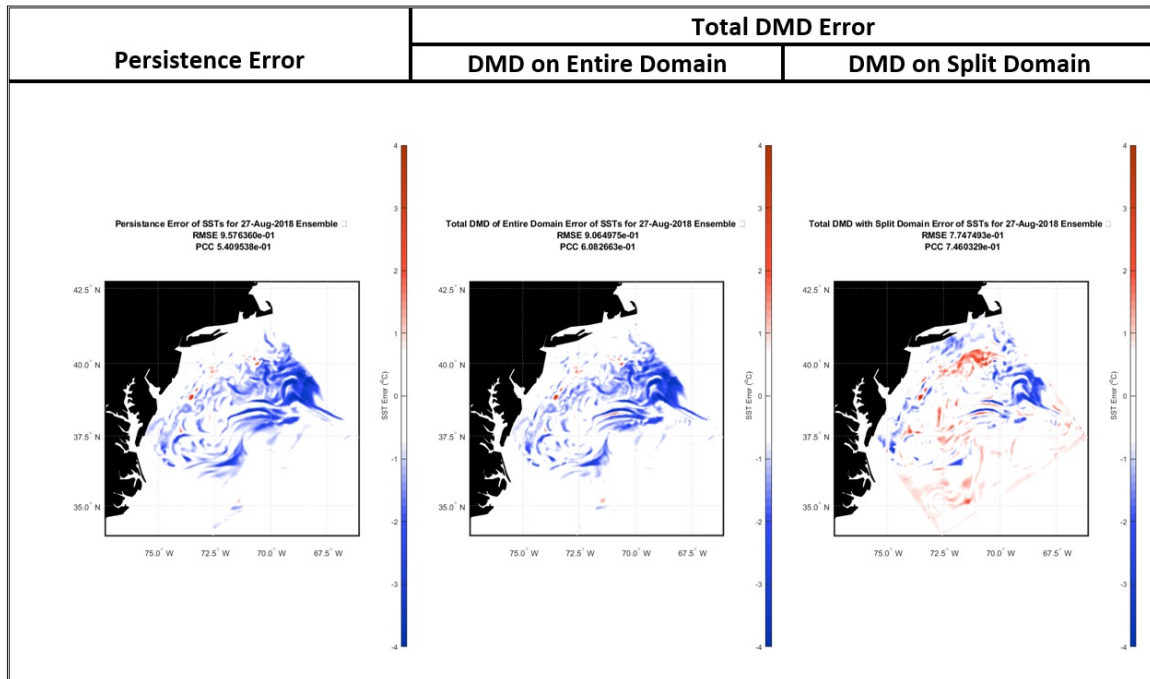


Figure 2-1: SST errors of 12 hour forecast for persistence (left), Total DMD applied to entire domain simultaneously (center), Total DMD on shelf/deep water separately (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment.

### 2.6.3 Training Size

We evaluated the effect of the size of the training set on the performance of DMD methods. To illustrate results, we show in Fig. 2-2 the zonal velocity errors from the persistence and exact DMD forecasts on September 16, 2006 19Z, for the AWACS/SW06 experiment. Here, for a 12-hr state forecast, persistence results in a PCC of 0.83; for 300 snapshot training, exact DMD results in a PCC of 0.87, and for 800 snapshot training, exact DMD results in a PCC of 0.91. In general, we find that the DMD methods provide better forecasts when the number of training snapshots increases at the expense of longer computation times.

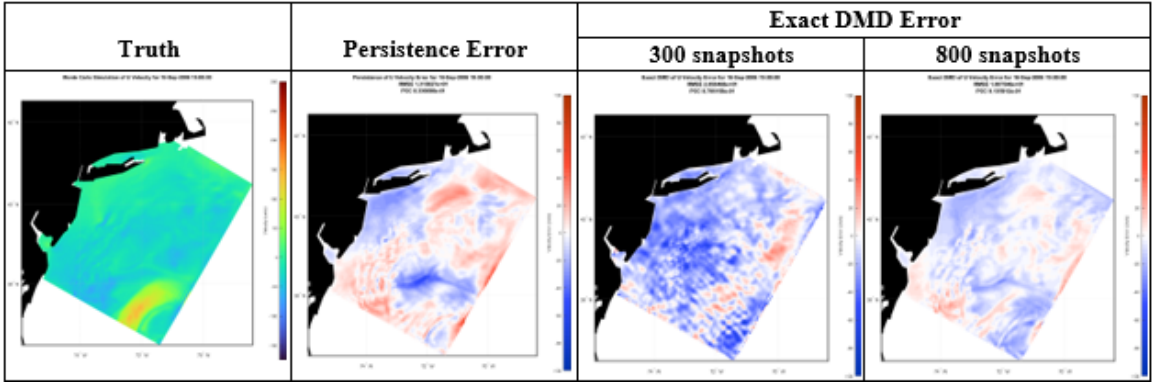


Figure 2-2: Zonal velocity truth (left), errors for the persistence (second from left) and exact DMD forecasts on Sep 16, 2006 19Z, for the AWACS/SW06 experiment, training with either 300 snapshots (second from right) or 800 snapshots (right).

Even though increasing number of training snapshots usually improves prediction skill, in most instances for the AWACS/SW06 experiment, around 100 to 200 hourly training snapshots allowed for reasonable predictions of about 3 days into the future for SST, Zonal Velocity, and Salinity. An 84 hour SST prediction error alongside persistence error using 199 training snapshots is shown in Fig. 2-3 to demonstrate this skill. In the case of the 84 hour forecast on Aug 25, 2006 19Z, the persistence forecast has a PCC of just 0.15 and the exact DMD forecast has a PCC 0.81. As would normally be expected, the root mean squared error is also lower for the DMD prediction (0.80 °C) compared to persistence (1.54 °C).

### 2.6.4 3D Forecasting

The DMD architecture allows for three-dimensional (3D) prediction provided that the 3D snapshots are arranged into column vectors  $\mathbf{x}_k$  and  $\mathbf{x}_{k+1}$ . In the following example, we used

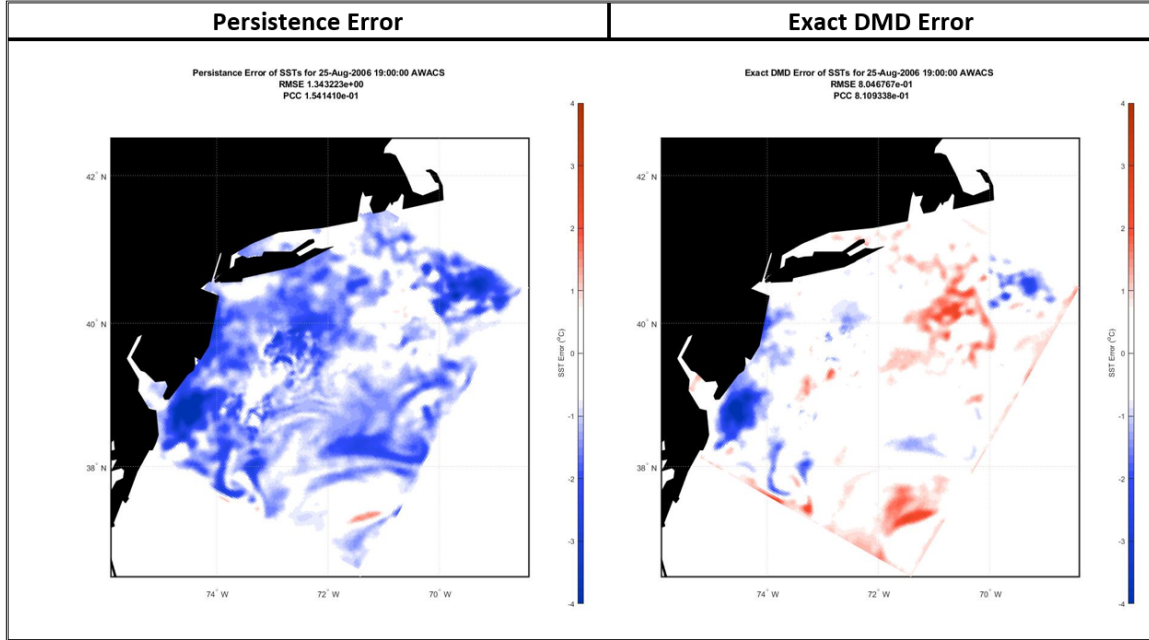


Figure 2-3: SST errors of 84 hour forecast for persistence (left) and Exact DMD (right) on Aug 25, 2006 19Z using 199 training snapshots (with 108 DMD modes used) from the AWACS/SW06 experiment.

100 training snapshots from the AWACS/SW06 experiment in order to predict zonal velocity for the next 48 hours using Streaming DMD. The PCC values at PE levels 1, 5, 15, and 20 of predictions every 4 hours are shown from Aug 22, 2006 07Z to Aug 24, 2006 07Z in Table 2.2. The DMD prediction significantly outperform the persistence for nearly all times and layers after time 0 (except for around 36 hours). 3D predictions of other variables (SST and Salinity) also outperformed the persistence forecast in the vast majority of instances over the same time period.

### 2.6.5 Ensemble Mean and Variance

We have shown some results from deterministic predictions using DMD methods. We now consider stochastic PE forecasts (consisting of 300 ensemble members) from the POSYDON-POINT experiment. Variations among the ensemble members are due to perturbations applied to the initial conditions (ICs) but also due to different tidal forcing parameters and atmospheric forcing fields used to force the different ensemble members [65].

We compare three DMD approaches to make probabilistic forecasts. For evaluation, we use the true mean and standard deviation of all POSYDON-POINT experiment PE ensembles at each grid-point at the specified times.

PCC of Zonal Velocity Forecast at Level Indicated										
	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD
Time(hr)	Sea Surface		Level 5		Level 10		Level 15		Level 20	
0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.07	0.93	-0.02	0.93	-0.15	0.92	-0.27	0.92	-0.39	0.93
8	-0.25	0.79	-0.32	0.82	-0.40	0.86	-0.45	0.89	-0.46	0.90
12	0.66	0.84	0.65	0.85	0.64	0.87	0.65	0.88	0.72	0.89
16	0.30	0.79	0.25	0.78	0.19	0.75	0.13	0.74	0.09	0.77
20	0.38	0.80	0.34	0.78	0.29	0.74	0.23	0.72	0.15	0.75
24	0.72	0.90	0.71	0.91	0.70	0.91	0.69	0.91	0.68	0.93
28	-0.14	0.54	-0.20	0.55	-0.27	0.58	-0.35	0.64	-0.36	0.73
32	0.13	0.76	0.08	0.71	0.01	0.66	-0.05	0.68	-0.06	0.68
36	0.88	0.82	0.87	0.80	0.87	0.80	0.89	0.81	0.91	0.83
40	0.14	0.66	0.08	0.64	0.00	0.65	-0.08	0.67	-0.18	0.72
44	-0.35	0.61	-0.40	0.62	-0.46	0.64	-0.54	0.67	-0.59	0.70
48	0.71	0.78	0.70	0.79	0.69	0.81	0.67	0.83	0.70	0.85

Table 2.2: Pattern correlation coefficients (PCC) of zonal velocity prediction vs. forecast time (from Aug 22, 2006 07Z to Aug 24, 2006 07Z) for the AWACS/SW06 experiment, at PE levels 1, 5, 15, and 20. DMD prediction was made using 3D zonal velocities and Streaming DMD method with 100 training snapshots.

The first approach is a brute force Monte-Carlo method where we perform DMD on each ensemble member individually to make DMD predictions. The mean and standard deviation of the DMD forecasts is then computed at each grid point. With this approach, our probabilistic forecast is the statistics (mean, variance, etc.) of the ensemble of DMD forecasts, i.e. the statistics of the DMD forecasts.

In order to improve over the cost of performing DMD on each ensemble, the second approach is a batch ensemble forecast [56, 57]. Here, we first compute the DMD prediction of ensemble member 1 at time  $t$  and the DMD prediction of member 2 at the same time. We then take the mean of these two predictions at each grid point. We then compute the DMD prediction of member 3 at the same time and determined the mean of the predictions of members 1, 2, and 3. We repeat this for member 4 and so on, until the running mean of the DMD prediction converged. This resulted in an estimate of the mean DMD prediction at each grid point. For the variance field forecast, we used Welford’s online variance algorithm [123] to compute the sample variance at each grid point. The cost of this method is much smaller than performing DMD on each ensemble individually (for the POSYDON-POINT SST PE ensemble this resulted in 41 of 300 ensemble members used).

The third approach first determines the mean and variance of the ensemble members at each grid-point and training time  $t_k$ . We then employ the DMD algorithm twice, once for



the mean field and once for the variance fields. In this approach, our probabilistic forecast is the DMD forecast of the mean, variance, etc., i.e. the DMD forecast of the statistics.

We illustrate the above three approaches using 85 training snapshots to predict the statistics of the SST field. We made a 12 hour forecast of the SST mean and SST variance. The SST mean error results are shown for August 27, 2018 00Z on Fig. 2-4. The resultant SST PCC values when compared to the true SST field mean are 0.51 for persistence and i) 0.75 for the mean of Total DMD forecasts of all members, ii) 0.71 for the mean of Total DMD forecasts using 41 members, and iii) 0.39 for the Total DMD forecast using the mean field as the input. Due to non-linearities, these results show that the DMD prediction using the mean field as input performed the worse followed by the mean persistence field. The mean SST Total DMD forecasts using all members and 41 members, however, performed similarly well (with the more expensive forecast using all members logically having a slight edge).

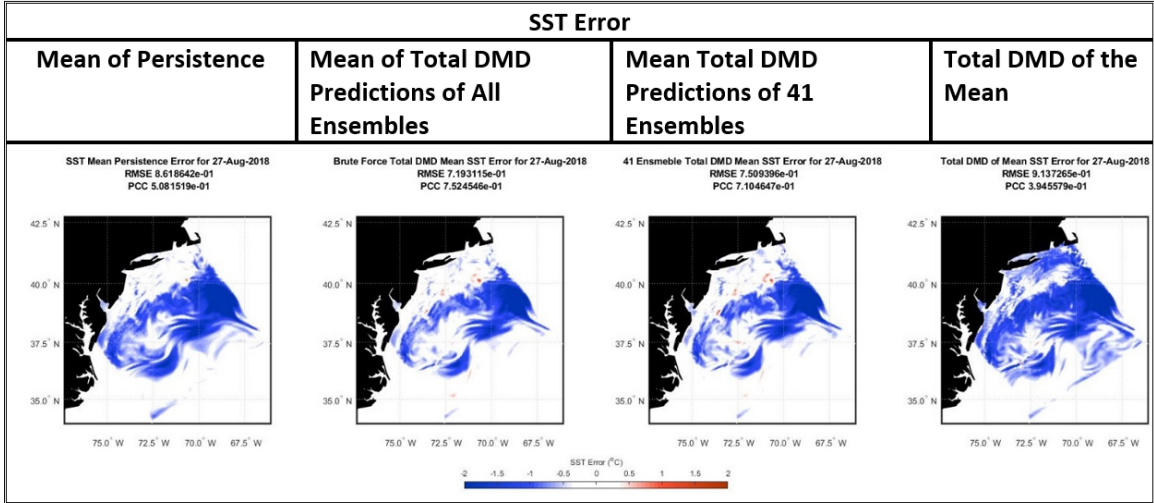


Figure 2-4: SST error from the true SST mean of 12 hour forecast for persistence (left), mean of Total DMD forecasts of all ensembles (second from left), mean of Total DMD forecasts of 41 ensembles (second from right), and Total DMD forecast using mean field (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment.

In Fig. 2-5, we show 12 hour future SST standard deviation fields for the truth (left), the Total DMD prediction using all members individually with a PCC of 0.80 (second from left), the 41 ensemble members Total DMD SST forecast with a PCC of 0.62 (second from right), and the Total DMD prediction using SST variance as the input with PCC of 0.82 (right). At first glance, the standard deviation fields appear similar. Upon further inspection, we see some areas (e.g. near the eastern most corner) where the 41 ensemble members Total DMD

SST forecast underestimates the standard deviation while the other two forecasts tend to slightly overestimate.

DMD with the mean of the ensemble as input performed poorly for the mean prediction compared to persistence but performed reasonably well for predicting the variance. The mean of DMD forecasts performed considerably better than the persistence mean but would require a high number of DMD forecasts to accurately predict the variance of the members.

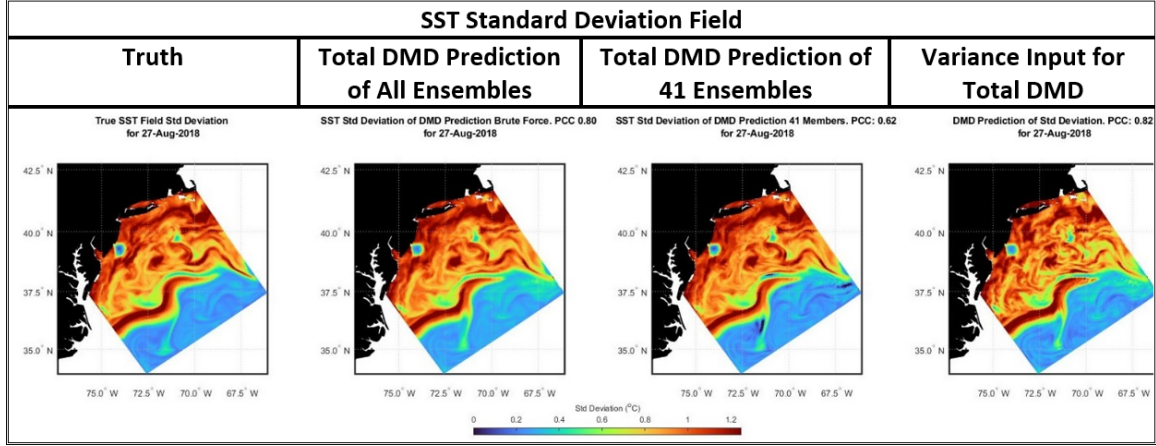


Figure 2-5: Standard deviation fields for true SST, the Total DMD SST forecasts of all ensemble members (PCC of 0.80), the 41 ensemble members Total DMD SST forecasts (PCC of 0.62), and the Total DMD forecast using variance as the input (PCC of 0.82) on August 27, 2018 00Z for the POSYDON-POINT experiment.

## 2.6.6 DMD of the Ensemble

As we mentioned earlier, we utilized another approach to combine ensemble members to make probabilistic DMD predictions [119]. With a total of  $N$  ensemble members and  $m$  times, each snapshot can be represented as  $\mathbf{x}_k^q$  where  $k$  is the snapshot time and  $q$  is the  $q$ -th ensemble member. DMD can then be applied to matrices

$$\mathbf{X} = \begin{bmatrix} | & & | & & | & & | & & | \\ \mathbf{x}_1^1 & \cdots & \mathbf{x}_{m-1}^1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_{m-1}^2 & \cdots & \mathbf{x}_1^N & \cdots & \mathbf{x}_{m-1}^N \\ | & & | & & | & & | & & | & \end{bmatrix} \quad (2.44)$$

and

$$\mathbf{X}' = \begin{bmatrix} | & & | & & | & & | & & | \\ \mathbf{x}_2^1 & \cdots & \mathbf{x}_m^1 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_m^2 & \cdots & \mathbf{x}_2^N & \cdots & \mathbf{x}_m^N \\ | & & | & & | & & | & & | & \end{bmatrix}. \quad (2.45)$$

This approach allows for extracting the DMD modes and eigenvalues from the entire ensemble set (or a specified number of ensemble members) simultaneously. With the overall DMD modes and eigenvalues, we can make prediction of a particular ensemble member using an initial condition (used to calculate the DMD amplitudes).

Using 60 ensemble members and 85 hour training data from the POSYDON-POINT experiment, we performed DMD on SST values as discussed above. We then predicted SST values for ensemble member 250 using the initial condition from that ensemble. The PCC results are listed in Table 2.3. The PCC values are generally better for the SSTs predicted using ensemble DMD modes compared to using exact DMD on the individual ensemble member. For the 12 hour forecast, we can see that persistence forecast of member 250 has a PCC of 0.48, Exact DMD forecast of member 250 has a PCC of 0.57, and Exact DMD forecast for member 250 using modes extracted from 60 members is 0.67.

SST PCC			
Time (hr)	Persistence	Exact DMD Using Only Member 250	Exact DMD using Ensemble DMD Modes
0	1.00	1.00	1.00
4	0.78	0.93	0.93
8	0.52	0.76	0.79
12	0.48	0.57	0.67

Table 2.3: PCCs of SST prediction vs. forecast time (Aug 26, 2018 12Z to Aug 27, 2018 00Z) for the POSYDON-POINT experiment ensemble member 250. Persistence forecast PCCs are consistently lower than Exact DMD forecasts. The DMD forecasts were made using DMD only on member 250 (center column) and using DMD modes extracted using 60 members (right column). Using the DMD modes extracted from multiple members provides better forecasts.

These results indicate that using multiple ensemble members can capture the underlying DMD better than just using a single member. Having the modes extracted using a portion the ensemble reduces storage and computation costs compared to using the full ensemble. This could prove advantageous for unmanned vehicles. The DMD mode could be computed off the vehicle, then a reduced number of DMD modes could be sent to the vehicle for relatively inexpensive predictions with reduced communication and storage needs.

## 2.6.7 Subspace Projections

We now illustrate the projections and compression of large ocean forecasts into pre-defined or adaptive subspaces. The projected forecasts, due to their much reduced dimension, can be readily transmitted to platforms with limited bandwidth. However, this reduction could affect accuracy. To show the effectiveness of POD modes for such reduction, we used 85 hour sea surface zonal velocity training data from ensemble member 150 of the POSYDON-POINT experiment for the period 23–27 August 2018 to determine the ‘past’ POD modes. We then projected the subsequent 12 hour PE forecast onto these past POD modes. The initial PE forecast was a vector with 54960 entries (440 kB), the projected PE forecast had just 22 entries (0.176 kB) using 90% variance-explained criterion. The remote platform, pre-loaded with these 22 past POD modes (less than 10 MB storage), would be able to reconstruct the forecast. In Fig. 2-6, we show the error between the truth full PE forecast and the 12 hour reconstructed POD-projected forecast next to 12 hour persistence forecast error for surface zonal velocity. PCC for the reconstructed projection is 0.81, while for persistence, it is 0.26. Results for a 24 hour forecast using 73 hour training data (not shown), had a PCC of 0.70 for reconstructed PE forecast onto past POD modes and a PCC of 0.48 for persistence forecast.

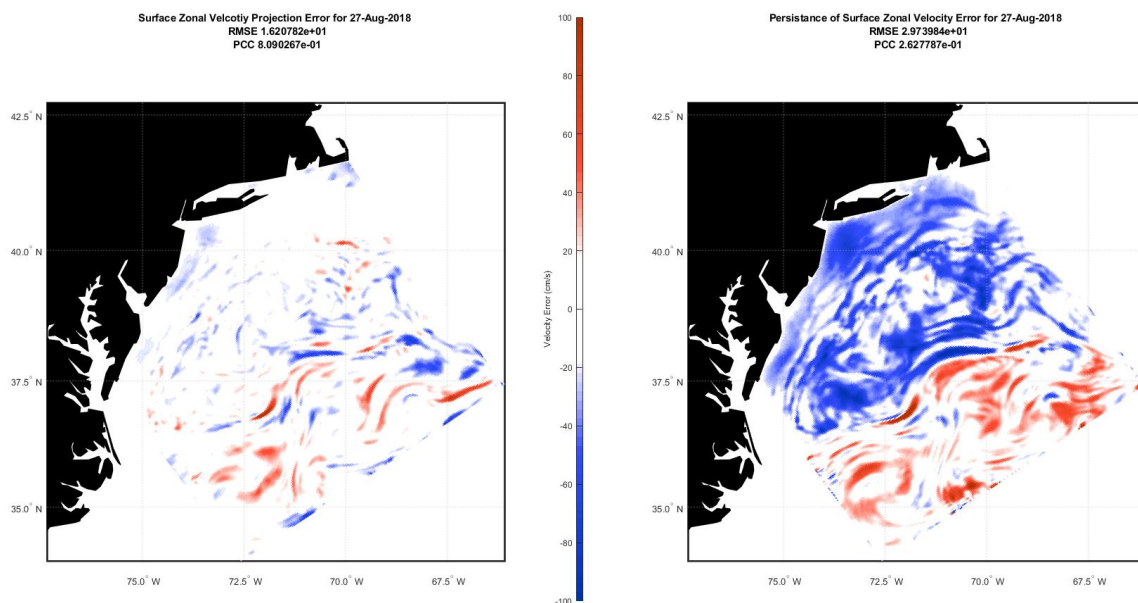


Figure 2-6: Surface Zonal Velocity errors for August 27, 2018 00Z for the POSYDON-POINT experiment. The reconstructed 12 hour PE forecast projection on past POD modes (left) with a PCC of 0.81 and the persistence forecast (right) with a PCC of 0.26.

### 2.6.8 Summary

Several reduced order modeling schemes for regional ocean forecasting onboard autonomous platforms at sea were described, investigated, and evaluated. We evaluated Dynamic Mode Decomposition (DMD) methods for ocean PE simulations by comparing several schemes including domain splitting, adjusting training size, and utilizing 3D inputs. Three different approaches that combine uncertainty with DMD were also investigated and found to be practical, especially if we employ either an ensemble of DMD forecasts or the DMD of an ensemble of forecasts. Projecting / compressing high-fidelity forecasts using schemes such as POD projection and K-SVD for sparse representation also showed promise for distributing forecasts to remote vehicles. In the following chapter, we introduce data assimilation techniques and how we integrate this with DMD modeling for improved stochastic predictions with data assimilation onboard.

THIS PAGE INTENTIONALLY LEFT BLANK

## Chapter 3

# Data Assimilation

### 3.1 Introduction

We introduce the field of data assimilation (DA) as a method to improve our forecasts using a limited amount of gappy measured data (either by the platform itself or communicated inexpensively to the platform from a reachback cell, separate manned platform, or another autonomous platform). Data assimilation is the process of optimally combining a numerical model with observations [28]. Often these observations are limited and thus data assimilation techniques allow for inexpensive methods to improve the model and its forecasts.

In our framework, an approximate dynamical model with uncertain estimates of initial and boundary conditions has been created forming an ensemble of simulations. This model is normally computationally intense and impractical to perform onboard the vehicle for forecasting. Instead, if we had an ensemble of simulations prior to deployment of an autonomous vehicle, we could use DMD to extract the underlying dynamics of the model to allow for predictions and creation of an ensemble of forecasts that are not required to be stored in their entirety (only need to have DMD modes, DMD eigenvalues, and DMD coefficients which can be much smaller than the full state ensemble for forecast duration period). Additionally, while the vehicle is in-situ we may have measurements (which we often refer to as data) collected at different space and time locations either by the vehicle, data communicated to the vehicle by local vessels/platforms, or a combination of those. The computation of the probability density function (pdf) of the model solutions conditioned on the measured data defines the data assimilation [28] considered in this thesis.

As mentioned in the Thesis Overview, we combine Dynamic Mode Decomposition with

Ensemble Kalman Filter (EnKF) and the Gaussian Mixture Model-Dynamically Orthogonal Filter (GMM-DO) as methods for stochastic field prediction. We start by describing EnKF and GMM-DO. We tend to focus more on GMM-DO as our DA scheme of choice because GMM-DO results tend to perform at or better than EnKF results. This occurs because when the optimum mixture component is one ( $M = 1$ ) the results of GMM-DO and EnKF are the same. On the other hand, if the optimum mixture component is greater than one ( $M > 1$ ), then the results of GMM-DO are more accurate than results of EnKF because the GMM-DO filter better generates a pdf that more accurately represents the true distribution.

The following sections define and describe EnKF and GMM-DO. It is important to note that we utilize both DA schemes in the stochastic subspace vice the full state space as a means to reduce the computational cost and make these schemes feasible for use onboard autonomous platforms with limited storage and computing capabilities.

Section 3.2 reviews techniques presented in [28], [98], and [66]. Section 3.3 reviews [109]. Section 3.4 presents our novel technique that integrates the DMD model with the GMM-DO filter.

## 3.2 Ensemble Kalman Filter

### 3.2.1 Description

In 1960 Kalman published a paper describing a filtering technique that was subsequently used to aid in the Apollo moon landing missions [45] [97]. Since this time, the Kalman filter (KF) has become a popular data filtering scheme that has been used in thousands of engineering applications.

The Ensemble Kalman Filter is a Monte Carlo implementation of a Bayesian update. With the prior (or forecast) probability density function (pdf) of the state of a modeled system and the likelihood, Bayes' theorem is used to obtain the posterior (or analysis) pdf after the likelihood has been taken into account. In our case the model we use to advance the posterior pdf is learned from past training inputs via DMD.

The EnKF represents the pdf of the system state using a collection of state vectors, called an ensemble, and replaces the covariance matrix used by KF by a sample covariance computed from the ensemble. The ensemble members are treated as if they are random, but the ensemble members are not truly independent. A significant advantage of the EnKF is



that advancing the pdf in time (using a model like DMD) can be done by advancing each ensemble member [28].

### 3.2.2 Formulation

The EnKF is a Monte Carlo approximation of the Kalman filter. The main advantage is EnKF avoids saving and evolving the full covariance matrix of the pdf of the state vector. Instead, the pdf is represented by the prior ensemble at time  $k$  for  $N$  total ensemble members

$$\mathbf{X}_k^f = \begin{bmatrix} | & | & & | \\ \mathbf{x}_k^1 & \mathbf{x}_k^2 & \cdots & \mathbf{x}_k^N \\ | & | & & | \end{bmatrix}. \quad (3.1)$$

$\mathbf{X}_k^f$  is a  $n \times N$  matrix whose columns are the ensemble members at time  $k$ . Despite the fact that the ensemble members (columns) are not truly independent they are deemed to be approximately independent and thus we proceed as if they are actually independent. Observation data  $\mathbf{d}$  is replicated into a  $N_{obs} \times N$  matrix (where  $N_{obs}$  is the number of observations collected at time  $k$ )

$$\mathbf{D} = \begin{bmatrix} | & | & & | \\ \mathbf{d}^1 & \mathbf{d}^2 & \cdots & \mathbf{d}^J \\ | & | & & | \end{bmatrix} = [\mathbf{d}^j], \mathbf{d}^j = \mathbf{d} + \epsilon^j, \epsilon^j \sim N(0, \mathbf{R}), \quad (3.2)$$

where  $\mathbf{R}$  is the covariance matrix that describes the estimate of the error of the observation data.

Each column of  $\mathbf{D}$  consists of the observation vector  $\mathbf{d}$  plus a random vector from the  $N_{obs}$ -dimensional normal distribution  $N(0, \mathbf{R})$ . As shown above, the columns of  $\mathbf{X}_k^f$  are a sample from the prior probability distribution, therefore the columns of

$$\hat{\mathbf{X}} = \mathbf{X}_k^f + \mathbf{K}(\mathbf{D} - \mathbf{H}\mathbf{X}_k^f) \quad (3.3)$$

form a sample from the posterior probability distribution where  $\mathbf{H}$  is the observation matrix that maps the observations to the state space and  $\mathbf{K}$  is the Kalman Gain matrix. The Kalman Gain matrix can be determined by using the sample covariance  $\mathbf{C}$  computed from the ensemble members (called the ensemble covariance) as follows:

$$\mathbf{K} = \mathbf{C}\mathbf{H}^T(\mathbf{H}\mathbf{C}\mathbf{H}^T + \mathbf{R})^{-1}. \quad (3.4)$$

The basic implementation from [28], allows us to determine the ensemble mean and sample covariance  $\mathbf{C}$  by the following:

$$E \left[ \mathbf{X}_k^f \right] = \frac{1}{N} \sum_{j=1}^N \mathbf{x}^j, \quad (3.5)$$

$$\mathbf{C} = \frac{\mathbf{B}\mathbf{B}^T}{N-1}, \quad (3.6)$$

where

$$\mathbf{B} = \mathbf{X}_k^f - E \left[ \mathbf{X}_k^f \right] \mathbf{e}_{1 \times N} = \mathbf{X}_k^f - \frac{1}{N} (\mathbf{X}_k^f \mathbf{e}_{N \times 1}) \mathbf{e}_{1 \times N} \quad (3.7)$$

and  $\mathbf{e}$  indicates the matrix or vector of all ones of specified dimensions.

It is now possible to determine the posterior ensemble

$$\mathbf{X}_k^a = \mathbf{X}_k^f + \mathbf{K}(\mathbf{D} - \mathbf{H}\mathbf{X}_k^f). \quad (3.8)$$

Conveniently, the covariance matrix  $\mathbf{R}$  is always positive semidefnite and usually positive definite, so an inverse in (3.4) exists and the Cholesky decomposition can be used to speed up computation [80].

### 3.2.3 Error Subspace Statistical Estimation

The EnKF is most often used in the full state space as described above. In the case where ensemble methods are used on autonomous platforms with limited storage and processing capabilities, this is often not practical or feasible. In his data assimilation via Error Subspace Statistical Estimation (ESSE) [62], Lermusiaux suggests further condensing the analysis presented by the Ensemble Kalman filter to a mere subspace of the error covariance matrix, thus focusing only on the dominant structures obtained through an appropriate orthonormal decomposition [62]. By limiting attention to this reduced subspace, he disregards less pronounced structures and consequently lessens the computational costs involved. This is the idea we have implemented to investigate feasibility for data assimilation on autonomous vehicles. Rather than using the sample covariance matrix as in the Ensemble Kalman filter, Lermusiaux proposes to retain only the subspace corresponding to its dominant rank- $s$

reduction, identified by use of the Singular Value Decomposition (SVD). By defining

$$\mathbf{M} = \{\mathbf{x}_k^j\} - \{\bar{\mathbf{x}}_k\} \quad (3.9)$$

where  $\bar{\mathbf{x}}_k$  is the mean of the forecast ensemble, he then proceeds by taking the SVD,

$$\text{SVD}_s[\mathbf{M}] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.10)$$

to obtain the  $s$  most dominant basis vectors,  $\mathbf{E}_s = \mathbf{U}$ , with associated eigenvalues  $\Lambda_s = \frac{1}{N-1}\mathbf{\Sigma}^2$ . This allows for an estimate for the error covariance matrix to be determined from which one proceeds with the Kalman update equation in the decomposed/subspace form. These efficiencies allowed Lermusiaux to demonstrate the first real-time ensemble data assimilation done at sea in the Strait of Sicily in 1996 utilizing ESSE [56].

### 3.3 GMM-DO Filter

The GMM-DO filter was originally introduced by Sondergaard and Lermusiaux [109] [110]. This data assimilation technique combines the use of Gaussian mixture models, the Expectation-Maximization (EM) algorithm and the Bayesian Information Criterion (BIC) to accurately approximate distributions based on Monte Carlo data allowing for fast and effective Bayesian inference. The GMM-DO filter couples these concepts with an efficient representation of the evolving probabilistic description of the uncertain dynamical field: the Dynamically Orthogonal field equations. By limiting attention to a dominant evolving stochastic subspace of the complete state space, we can determine the GMM distributions much more quickly and efficiently compared to working in the state space. In the following we describe the fundamental concepts used for the GMM-DO filter including Gaussian mixture models, the Expectation-Maximization algorithm, the Bayesian Information Criterion, and the Dynamically Orthogonal Field equations. Most of the following discussion of GMM-DO and related concepts comes from [108] [109].

### 3.3.1 Gaussian Mixture Models

The probability density function (pdf) for a random vector  $\mathbf{X} \in \mathbb{R}^n$  that is distributed according to a multivariate GMM is

$$p_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^M \pi_j \times \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j, \mathbf{P}_j) \quad (3.11)$$

subject to

$$\sum_{j=1}^M \pi_j = 1. \quad (3.12)$$

We define the following terms as the mixture complexity (or number of mixture components)  $M \in \mathbb{N}$ , the mixture weights  $\pi_j \in [0, 1]$ , the mixture mean vectors  $\bar{\mathbf{x}}_j \in \mathbb{R}^n$ , and the mixture covariance matrices  $\mathbf{P}_j \in \mathbb{R}^{n \times n}$ . For ease of reference table 3.1 lists the notation used for GMM, EM, BIC, and the GMM-DO framework. Note that this notation may differ in certain areas from [109] and [110] to match the DMD framework we have already presented. The multivariate Gaussian density function can given by:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{x}})\right]. \quad (3.13)$$

GMMs provide a semi-parametric framework in which to approximate unknown distributions [83]. GMMs can essentially be looked at as a flexible alternative to the fully parametric Gaussian distribution where  $M = 1$  and the kernel density estimator [107] where  $M = N$  (the number of data points/ensemble members). The fully parametric Gaussian model often forces the data into a structure that is not true to real life and is incapable for modeling multi-modal or largely skewed inputs. The kernel density estimator requires the retention of all  $N$  members for inference, a computationally expensive task. Additionally, due to the over fitting that is associated with fitting a kernel to every member, kernel density estimators often necessitate heuristic choosing of the kernel's shape parameter. For these reasons, GMMs are popular due to their efficiency and accuracy at representing complex distributions. For large complexity and small covariance, as a matter of fact, a GMM model converges uniformly to smooth distributions [2]. An example of the three distribution types described above (from [109]) is shown in Fig. 3-1.

Some other methods of approximating arbitrary probability distributions previously con-

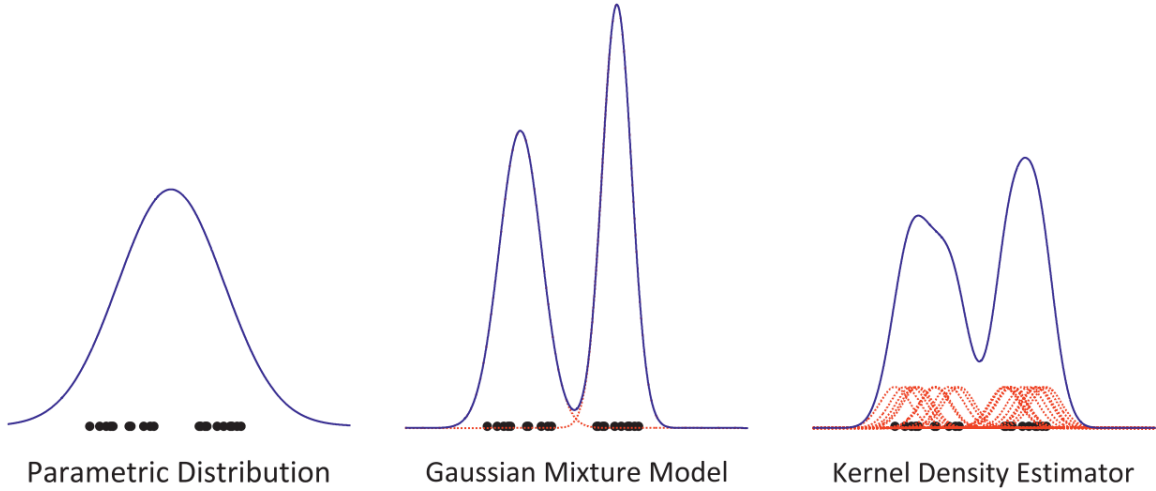


Figure 3-1: Parametric (Gaussian) distribution, GMM, and Kernel Density approximation of 20 samples generated from mixture of uniform distributions:  $p_X(x) = \frac{1}{2} \mathcal{U}(x; -8, -1) + \frac{1}{2} \mathcal{U}(x; 1, 8)$  where  $\mathcal{U}(x; a, b) = \frac{1}{b-a}$  is the continuous uniform probability density function for random variable  $X$ . Source [109].

sidered suffered from drawbacks. For example, the Gram-Charlier expansion and the Edgeworth expansion [2] are not valid when truncated because they must integrate to be one and positive everywhere. Another example discussed in [2], the Pearson-type density function, does not work well with Bayesian inference. On the other hand, Gaussian mixture models are shown by equations (3.11), (3.12), and (3.13) to be valid and notably for Gaussian observation models, they turn the Bayesian update trivial by using the concept of conjugacy. Conjugacy is defined in [15] as:

**Definition: Conjugate Prior**

*Let  $\mathcal{F}$  denote the class of probability densities  $p_{Y|X}(y|x)$ . A class  $\mathcal{G}$  of prior distributions on  $X$ ,  $p_X(x)$ , is a conjugate family for  $\mathcal{F}$  if the posterior distribution for  $X$  given  $Y$  via Baye's Law, is also in the class  $\mathcal{G}$  for all  $p_{Y|X} \in \mathcal{F}$ , all  $p_X(x) \in \mathcal{G}$ , and all  $y \in Y$ .*

This definition is for the univariate random variable case, but can be extended to the multivariate random vector case through a linear operator (like  $\mathbf{Y} = \mathbf{H}\mathbf{X}$ ).

### 3.3.2 Expectation-Maximization Algorithm

The following description of the Expectation-Maximization (EM) algorithm is predominantly based on [81], [82], and [83].

The EM algorithm describes an iterative procedure for estimating the parameters of a

target distribution that maximize the probability of obtaining the available inputs,  $\{\mathbf{x}\} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , thus arriving at the Maximum Likelihood (ML) estimate for the unknown set of parameters. The nice thing about ML estimators are that they are consistent and asymptotically efficient [6].

For many realistic cases, obtaining the ML estimate by differentiating the parametric probability distribution with respect to a parameter of interest and equating to zero, gives a nonlinear result that lacks closed form solution [106]. These cases mean that the use of numerical optimization methods is best.

The EM algorithm is often introduced, in the literature, as a method to estimate ML parameters when incomplete inputs or data is present. One crucial step is the *completion* of the data (by imputing from known data like techniques presented in [120]). This data completion allows for the ML solution to be much more tractable and computationally efficient. The completed data problem normally allows for a closed form solution to the estimation problem and allows us to obtain the ML estimation parameters by a simple partial differential. The following contains the EM algorithm including some of the important concepts already discussed.

Following [106], we let  $\{x\} = \{x_1, \dots, x_N\}^T$  be a set of available input,  $\{z\}$  the *complete* input vector, and  $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_M\}^T$  be the set of parameters (which are to be computed) of the chosen distribution  $p_{\{z\}}(\{z\}; \boldsymbol{\theta})$ . We assume that the input is a unique and deterministic function of the complete input, for example  $\{x\} = g(\{z\})$ . Using the Total Probability Theorem [6], we can write:

$$p_{\{z\}}(\{z\}; \boldsymbol{\theta}) = \sum_{\{x\}} p_{\{z\}|\{x\}}(\{z\}|\{x\}; \boldsymbol{\theta}) \times p_{\{x\}}(\{x\}; \boldsymbol{\theta}) \quad (3.14)$$

$$= p_{\{z\}|\{x\}}(\{z\}|g(\{z\}); \boldsymbol{\theta}) \times p_{\{x\}}(g(\{z\}); \boldsymbol{\theta}). \quad (3.15)$$

By taking the natural logarithms, we consequently obtain for any value of  $\{z\}$  that satisfies  $\{x\} = g(\{z\})$ :

$$= \log(p_{\{x\}}(\{x\}; \boldsymbol{\theta})) = \log(p_{\{z\}}(\{z\}; \boldsymbol{\theta})) - \log(p_{\{z\}|\{x\}}(\{z\}|\{x\}; \boldsymbol{\theta})). \quad (3.16)$$

Now taking expectation with respect to the *complete* inputs, which were conditioned based on available input and parameterized by arbitrary vector  $\tilde{\boldsymbol{\theta}}$  (which we are aiming to opti-

mize):

$$E[(\cdot)|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}}] = \int_{\{z\}} (\cdot) p_{\{Z\}|\{X\}}(\{z\}|\{x\}; \tilde{\boldsymbol{\theta}}) d\{z\}, \quad (3.17)$$

and using (3.16) with (3.17) we obtain

$$\log(p_{\{X\}}(\{x\}; \boldsymbol{\theta})) = E[\log(p_{\{Z\}}(\{z\}; \boldsymbol{\theta})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}}) - E[\log(p_{\{Z\}|\{X\}}(\{z\}|\{x\}; \boldsymbol{\theta})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}})]. \quad (3.18)$$

For convenience, we define

$$U(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = E[\log(p_{\{Z\}}(\{z\}; \boldsymbol{\theta})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}})] \quad (3.19)$$

$$V(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = -E[\log(p_{\{Z\}|\{X\}}(\{z\}|\{x\}; \boldsymbol{\theta})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}})] \quad (3.20)$$

to simplify (3.18) and obtain the expression

$$\log(p_{\{X\}}(\{x\}; \boldsymbol{\theta})) = U(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) + V(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}). \quad (3.21)$$

We then apply the of Gibbs' inequality [79] and see that

$$V(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) = -E[\log(p_{\{Z\}|\{X\}}(\{z\}|\{x\}; \boldsymbol{\theta})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}})] \quad (3.22)$$

$$\geq -E[\log(p_{\{Z\}|\{X\}}(\{z\}|\{x\}; \tilde{\boldsymbol{\theta}})|\{X\} = \{x\}; \tilde{\boldsymbol{\theta}})] \quad (3.23)$$

$$\equiv V(\tilde{\boldsymbol{\theta}}; \tilde{\boldsymbol{\theta}}). \quad (3.24)$$

If  $\tilde{\boldsymbol{\theta}}$  is denoted the present estimate for the parameter vector, by choosing  $\boldsymbol{\theta} \neq \tilde{\boldsymbol{\theta}}$  such that  $U(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) \geq U(\tilde{\boldsymbol{\theta}}; \tilde{\boldsymbol{\theta}})$  is satisfied, it guarantees that

$$\log(p_{\{X\}}(\{x\}; \boldsymbol{\theta})) = U(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) + V(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}) \quad (3.25)$$

$$\geq U(\tilde{\boldsymbol{\theta}}; \tilde{\boldsymbol{\theta}}) + V(\tilde{\boldsymbol{\theta}}; \tilde{\boldsymbol{\theta}}) \quad (3.26)$$

$$\log(p_{\{X\}}(\{x\}; \tilde{\boldsymbol{\theta}})) \quad (3.27)$$

Upon iterations, the estimate for the parameter vector monotonically increases the log likelihood of generating the data at hand. Assuming further that the likelihood is bounded from above, we are thus guaranteed to converge to a stationary point and as such obtain an

estimate for the ML parameter vector [15].

We summarize the definition of the EM algorithm in the following:

**Definition: The Expectation Maximization Algorithm**

*Given available inputs,  $\{x\} = \{x_1, \dots, x_N\}$ , initial parameter estimate  $\boldsymbol{\theta}^{(0)}$ , proposed complete inputs  $\{z\}$  with user-specified distribution,  $p_{\{Z\}}(\{z\}; \boldsymbol{\theta})$ , repeat until convergence:*

- *Using the present parameter estimate,  $\boldsymbol{\theta}^{(k)}$ , for the following*

$$U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}) = E[\log(p_{\{Z\}}(\{z\}; \boldsymbol{\theta} | \{X\} = \{x\}; \boldsymbol{\theta}^{(k)})]. \quad (3.28)$$

- *Estimate for the parameter vector,  $\boldsymbol{\theta}^{(k+1)}$ , is updated by maximizing  $U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$ :*

$$\boldsymbol{\theta}^{(k+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}(U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})). \quad (3.29)$$

Due to the fact that we utilize the EM algorithm in conjunction with multivariate GMMs, we explain the use of of EM with GMMs in the following.

### 3.3.3 The EM algorithm used with Gaussian Mixture Models

We assume that we have an a set of inputs  $\{\mathbf{x}\} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , produced by the multivariate Gaussian distribution:

$$p_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^M \pi_j \times \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j, \mathbf{P}_j) \quad (3.30)$$

for which we assume the ML estimate for the parameter vector

$$\boldsymbol{\theta} = \{\pi_1, \dots, \pi_M, \bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_M, \mathbf{P}_1, \dots, \mathbf{P}_M\}. \quad (3.31)$$

Note the convenient notation of allowing the parameter vector to contain both non-transposed vectors as well as full matrices for the purpose of this thesis. For now we assume that the mixture complexity,  $M$ , is known and fixed but subsequently we will demonstrate that the Bayesian Inference Criterion allows for the optimal choice of mixture complexity. We augment the inputs as needed to form the *complete* data set

$$\{\mathbf{z}\} = \{\mathbf{c}_1, \mathbf{x}_1, \dots, \mathbf{c}_N, \mathbf{x}_N\}, \quad (3.32)$$



where  $\mathbf{c}_i$  is an indicator vector such that

$$(\mathbf{c}_i)_j = \begin{cases} 1, & \text{if data point } \mathbf{x}_i \text{ was generated by mixture } j, \\ 0, & \text{otherwise,} \end{cases} \quad (3.33)$$

with  $(\mathbf{c}_i)_j$  being the  $j^{\text{th}}$  element of vector  $\mathbf{c}_i$ .

By the assumed independence of the inputs, the probability distribution of the complete inputs takes the following form:

$$p(\{\mathbf{z}\}; \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{c}_i, \mathbf{x}_i; \boldsymbol{\theta}) \quad (3.34)$$

$$= \prod_{i=1}^N \prod_{j=1}^M [\pi_j \times \mathcal{N}(\mathbf{x}_i; \bar{\mathbf{x}}_j, \mathbf{P}_j)]^{(\mathbf{c}_i)_j}. \quad (3.35)$$

Taking the natural logarithms we obtain

$$\log[p(\{\mathbf{z}\}; \boldsymbol{\theta})] = \sum_{i=1}^N \sum_{j=1}^M (\mathbf{c}_i)_j \times [\log(\pi_j) + \log(\mathcal{N}(\mathbf{x}_i; \bar{\mathbf{x}}_j, \mathbf{P}_j))]. \quad (3.36)$$

In addition, we can take the conditional expectation of (3.36) with respect to the available inputs, arbitrarily parameterized by vector  $\boldsymbol{\theta}^{(k)}$  to get the expression to be maximized by the EM algorithm:

$$U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}) = E[\log\{p(\{\mathbf{z}\}; \boldsymbol{\theta}) | \{\mathbf{x}\}; \boldsymbol{\theta}^{(k)}\}] \quad (3.37)$$

$$\sum_{i=1}^N \sum_{j=1}^M E[(\mathbf{c}_i)_j | \{\mathbf{x}\}; \boldsymbol{\theta}^{(k)}] \times [\log(\pi_j) + \log(\mathcal{N}(\mathbf{x}_i; \bar{\mathbf{x}}_j, \mathbf{P}_j))] \quad (3.38)$$

For convenience, we define the following:

$$\tau_j(\mathbf{x}_i; \boldsymbol{\theta}^{(k)}) \equiv E[(\mathbf{c}_i)_j | \{\mathbf{x}\}; \boldsymbol{\theta}^{(k)}] \quad (3.39)$$

$$= \frac{\pi_j^{(k)} \times \mathcal{N}(\mathbf{x}_i; \bar{\mathbf{x}}_j^{(k)}, \mathbf{P}_j^{(k)})}{\sum_{m=1}^M \pi_m^{(k)} \times \mathcal{N}(\mathbf{x}_i; \bar{\mathbf{x}}_m^{(k)}, \mathbf{P}_m^{(k)})} \quad (3.40)$$

This completes the expectation step of the EM algorithm. We continue to the maximization step which determines the posterior parameter vector,  $\boldsymbol{\theta}^{(k+1)}$ , which maximizes  $U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$ . We augment  $U(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$  using Lagrange multipliers to determine updated mixture weights

$\pi_j^{(k+1)}$ . We thus introduce auxiliary function  $\Lambda$  with multiplier  $\lambda$ :

$$\Lambda = \sum_{j=1}^M \sum_{i=1}^N \tau_j(\mathbf{x}_i; \boldsymbol{\theta}^{(k)}) \times [\log(\pi_j) - \frac{1}{2} \log |\mathbf{P}_j| - \frac{1}{2} (\mathbf{x}_i - \bar{\mathbf{x}}_j)^T \mathbf{P}_j^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_j)] + \lambda \times (\sum_{k=1}^M \pi_k - 1). \quad (3.41)$$

By setting the gradient to zero with respect to  $\pi_p$  and  $\lambda$ , after solving and manipulating we obtain:

$$\pi_p^{(k+1)} = \frac{\sum_{i=1}^N \tau_p(\mathbf{x}_i; \boldsymbol{\theta}^{(k)})}{N} = \frac{N_p^{(k)}}{N} \quad (3.42)$$

where  $N_p^{(k)}$  represents the sum total of particles associated with a given mixture component  $p$  using the present estimated parameter  $\boldsymbol{\theta}^{(k)}$ . Now we can determine the unconstrained parameters,  $\bar{\mathbf{x}}_p^{(k+1)}$  and  $\mathbf{P}_p^{(k+1)}$ . The updated component mixture mean vectors are determined by setting the partial derivative to zero as follows:

$$\frac{\partial \Lambda}{\partial \bar{\mathbf{x}}_p} = 0 \quad (3.43)$$

and solving we obtain:

$$\bar{\mathbf{x}}_p^{(k+1)} = \frac{1}{N_p^{(k)}} \sum_{i=1}^N \tau_p(\mathbf{x}_i; \boldsymbol{\theta}^{(k)}) \times \mathbf{x}_i. \quad (3.44)$$

To obtain the updated mixture covariance matrices we solve

$$\frac{\partial \Lambda}{\partial \mathbf{P}_p} = 0 \quad (3.45)$$

to get

$$\mathbf{P}_p^{(k+1)} = \frac{1}{N_p^{(k)}} \sum_{i=1}^N \tau_p(\mathbf{x}_i; \boldsymbol{\theta}^{(k)}) \times (\mathbf{x}_i - \bar{\mathbf{x}}_p^{(k+1)})(\mathbf{x}_i - \bar{\mathbf{x}}_p^{(k+1)})^T. \quad (3.46)$$

This is the explanation of EM applied to GMMs. For the complete derivation refer to [108].

### 3.3.4 Bayesian Inference Criterion

Prior to this section the mixture complexity,  $M$ , was assumed known and constant. In reality the optimal mixture complexity is often unknown. Determining the best complexity of a GMM can be complicated and there are several methods that exist [27] [83] [24], here we focus and discuss the Bayesian Inference Criterion (BIC).

To start we assume that the parameter vector,  $\boldsymbol{\theta}$ , is random and that the mixture

complexity,  $M$ , at a particular discrete time is constant but unknown. To denote the prior distribution for  $\boldsymbol{\theta}$  for a given  $M$  we use  $p_{\boldsymbol{\theta}}(\boldsymbol{\theta}; M)$  and to denote distribution ensemble set conditioned on  $\boldsymbol{\theta}$  for a give  $M$  we use  $p_{\{\mathbf{x}\}|\boldsymbol{\theta}}(\{\mathbf{x}\}|\boldsymbol{\theta}; M)$ . In this thesis this latter distribution is the GMM.

Our goal is to determine a model complexity  $M$  that maximizes the likelihood of obtaining  $\{\mathbf{x}\}$ . Essentially by the assumed independence of the realizations, we are searching for the  $M$  that maximizes

$$p_{\{\mathbf{x}\}}(\{\mathbf{x}\}; M) = \prod_{i=1}^N p_{\mathbf{x}_i}(\mathbf{x}_i; M). \quad (3.47)$$

Effectively, Laplace's approximation is applied to the left-hand side of Bayes's law,

$$p_{\boldsymbol{\theta}|\{\mathbf{x}\}}(\boldsymbol{\theta}|\{\mathbf{x}\}; M) = \frac{p_{\{\mathbf{x}\}|\boldsymbol{\theta}}(\{\mathbf{x}\}|\boldsymbol{\theta}; M) p_{\boldsymbol{\theta}}(\boldsymbol{\theta}; M)}{p_{\{\mathbf{x}\}}(\{\mathbf{x}\}; M)}, \quad (3.48)$$

evaluated at the ML estimate for parameter vector  $\boldsymbol{\theta}$ . This allows us to arrive to

$$\frac{1}{N} L_{\mathbf{X}}^N(M) = \frac{1}{N} L_{\mathbf{X}}^N(\hat{\boldsymbol{\theta}}_{ML}, M) + \frac{1}{N} \log p_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}_{ML}; M) + \frac{K_M}{2N} \log(2\pi) - \frac{K_M}{2N} \log(N) - \frac{1}{N} \log \left| \mathbf{J}_{\mathbf{X}}(\hat{\boldsymbol{\theta}}_{ML}) \right|, \quad (3.49)$$

where  $K_M$  is the length of the parameter vector and  $\mathbf{J}_{\mathbf{X}}(\hat{\boldsymbol{\theta}}_{ML})$  is the expected Fisher information in any one realization  $\mathbf{x}_i$  evaluated at the ML estimate for the parameter vector  $\boldsymbol{\theta}$ . Additionally, the log likelihoods are defined as:

$$L_{\mathbf{X}}^N(M) = \sum_{i=1}^N \log(p_{\mathbf{x}_i}(\mathbf{x}_i; M)) \quad (3.50)$$

and

$$L_{\mathbf{X}}^N(\hat{\boldsymbol{\theta}}_{ML}, M) = \sum_{i=1}^N \log(p_{\mathbf{x}_i|\boldsymbol{\theta}}(\mathbf{x}_i|\hat{\boldsymbol{\theta}}_{ML}; M)). \quad (3.51)$$

When the number of ensemble members,  $N$ , is large we keep only the order one terms of (3.49) to determine the BIC:

$$BIC = \min_M \{-2L_{\mathbf{X}}^N(M)\} \quad (3.52)$$

$$\approx \min_M \{K_M \log(N) - 2L_{\mathbf{X}}^N(\hat{\boldsymbol{\theta}}_{ML}, M)\}, \quad (3.53)$$

where  $M$  is the number of realizations/ensemble members,  $M$  is the mixture complexity,

$L_{\mathbf{X}}^N(M)$  is the log-likelihood of the ensemble set integrated across all possible parameter values,  $L_{\mathbf{X}}^N(\hat{\boldsymbol{\theta}}_{ML}, M)$  is the log-likelihood of the ensemble set evaluated at the ML estimate for the parameter vector, and finally  $K_M$  is the number of parameters. The complexity  $M$  is chosen such that the BIC is minimized.

### 3.3.5 Dynamically Orthogonal Field Equations

To employ the discussed GMM with EM and BIC efficiently we now utilize the DO equations. The Dynamically Orthogonal (DO) Field Equations [103] [102] are a closed reduced set of evolution equations for general stochastic continuation fields,  $\mathbf{X}(\mathbf{r}, t; \omega)$ , described by a stochastic partial differential equation (SPDE):

$$\frac{\partial \mathbf{X}(\mathbf{r}, t; \omega)}{\partial t} = \mathcal{L}[\mathbf{X}(\mathbf{r}, t; \omega)] \quad (3.54)$$

with initial conditions (ICs)

$$\mathbf{X}(\mathbf{r}, t_0; \omega) = \mathbf{X}_0(\mathbf{r}, \omega) \quad (3.55)$$

and boundary conditions (BCs)

$$\mathcal{B}[\mathbf{X}(\mathbf{r}, t; \omega)]|_{\mathbf{r}=\boldsymbol{\xi}} = h(\boldsymbol{\xi}, t; \omega), \quad (3.56)$$

where  $\mathbf{r}$  denotes the position in space,  $t$  is time,  $\omega$  is a random event,  $\mathcal{L}[\cdot]$  is a general (could be nonlinear) differential operator,  $\mathcal{B}$  is a linear differential operator, and  $\boldsymbol{\xi}$  is the spatial coordinate denoting the boundary. Of note, there are two important assumptions made in the derivation of the DO equations. The first is that a generalized, time-dependent Karhunen-Loeve decomposition of the fields is used [64] [103]:

$$\mathbf{X}(\mathbf{r}, t; \omega) = \bar{\mathbf{x}}(\mathbf{r}, t) + \sum_{i=1}^{s(t)} \tilde{\mathbf{x}}_i \boldsymbol{\Psi}_i(t; \omega), \quad (3.57)$$

where  $\bar{\mathbf{x}}(\mathbf{r}, t) = E[\mathbf{X}(\mathbf{r}, t; \omega)]$  are the mean fields over  $\omega$ ,  $\tilde{\mathbf{x}}_i(\mathbf{r}, t)$  are orthonormal modes spanning the time-dependent stochastic subspace, and  $\boldsymbol{\Psi}_i(t; \omega)$  are zero-mean stochastic coefficients. In the following, the dimension of the subspace,  $s$ , changes with time but for notation simplicity we omit the  $t$  next to the  $s$ . The second important assumption is after the insertion of (3.54) into (3.56), a DO condition is imposed (the rate of change of the

subspace basis is orthogonal to itself over the physical domain)

$$\left\langle \frac{\partial \tilde{\mathbf{x}}_i(\cdot, t)}{\partial t}, \tilde{\mathbf{x}}_j(\cdot, t) \right\rangle = 0 \quad \forall \quad i, j \in \{1, \dots, s\}. \quad (3.58)$$

This allows for the original SPDE to be reduced to a set of DO equations consisting of a PDE for the evolution of the full state mean field  $\bar{\mathbf{x}}(\mathbf{r}, t)$ :

$$\frac{\partial \bar{\mathbf{x}}(\mathbf{r}, t)}{\partial t} = E[\mathcal{L}[\mathbf{X}(\mathbf{r}, t; \omega); \omega]], \quad (3.59)$$

a family of  $s$  PDEs for evolving the orthonormal modes  $\tilde{\mathbf{x}}_i(\mathbf{r}, t)$  that define the basis for the time-dependent stochastic subspace:

$$\frac{\partial \tilde{\mathbf{x}}_i(\mathbf{r}, t)}{\partial t} = \mathbf{\Pi}^\perp(E[\mathcal{L}[\mathbf{X}(\mathbf{r}, t; \omega); \omega]\psi_j(t; \omega)])C_{\Psi_i(t)\Psi_j(t)}^{-1}, \quad (3.60)$$

and a system of  $s$  stochastic differential equations for the coefficients  $\Psi_i(t; \omega)$  that define how the stochastic nature evolves with in the subspace

$$\frac{d\Psi_i(t; \omega)}{dt} = \langle \mathcal{L}[\mathbf{X}(\cdot, t; \omega); \omega] - E[\mathcal{L}[\mathbf{X}(\cdot, t; \omega); \omega]], \tilde{\mathbf{x}}_i(\cdot, t) \rangle, \quad (3.61)$$

where

$$\mathbf{\Pi}^\perp[F(\mathbf{r})] \equiv F(\mathbf{r}) - \langle F(\cdot), \tilde{\mathbf{x}}_k(\cdot, t) \rangle \tilde{\mathbf{x}}_k(\mathbf{r}, t) \quad (3.62)$$

is the projection of  $F(\mathbf{r})$  onto the null space of the stochastic subspace and

$$C_{\Psi_i(t)\Psi_j(t)}^{-1} \equiv E[\Psi_i(t; \omega)\Psi_j(t; \omega)] \quad (3.63)$$

is the correlation between random variables  $\Psi_i(t; \omega)$  and  $\Psi_j(t; \omega)$  used in (3.60).

The boundary conditions are:

$$\mathcal{B}[\bar{\mathbf{x}}(\mathbf{r}, t)]|_{\mathbf{r}=\boldsymbol{\xi}} = E[h(\boldsymbol{\xi}, t; \omega)], \quad (3.64)$$

$$\mathcal{B}[\tilde{\mathbf{x}}_i(\mathbf{r}, t)]|_{\mathbf{r}=\boldsymbol{\xi}} = E[h(\boldsymbol{\xi}, t; \omega)\Psi_j(t; \omega)]C_{\Psi_i(t)\Psi_j(t)}^{-1}, \quad (3.65)$$

and the initial conditions are:

$$\bar{\mathbf{x}}(\mathbf{r}, t_0) = \bar{\mathbf{x}}_0(\mathbf{r}) = E[\mathbf{X}_0(\mathbf{r}; \omega)], \quad (3.66)$$

$$\tilde{\mathbf{x}}_i(\mathbf{r}, t_0) = \tilde{\mathbf{x}}_{i0}(\mathbf{r}), \quad (3.67)$$

and

$$\Psi_i(t_0; \omega) = \langle \mathbf{X}_0(\cdot; \omega) - \bar{\mathbf{x}}_0(\cdot), \tilde{\mathbf{x}}_{i0}(\cdot) \rangle, \quad (3.68)$$

where  $i = 1, \dots, s$  and  $\tilde{\mathbf{x}}_{i0}(\mathbf{r})$  are orthonormal modes for the stochastic subspace at  $t_0$ .

The DO equations allow for the stochastic subspace and the stochastic coefficients to be dynamically evolved in time after being initialized based on the initial pdf and evolved according to SPDE governing  $\mathbf{X}(\mathbf{r}, t; \omega)$  and its boundary conditions. This presents a significant advantage compared to the proper orthogonal decomposition (POD) [93] [42] and polynomial chaos [34], which both fix in time parts of their truncated expansion.

### 3.4 DMD with GMM-DO Filter Description

For the purpose of this thesis, we replace the DO evolution in time with a different evolution in time, namely the DMD forecast. The following is a brief description of DMD with the GMM-DO filter. Fig. 3-2 shows a DMD with GMM-DO filter flow chart for reference. For a more detailed description of GMM-DO including comparisons and discussions of similar methods see [108]. Note that the discussion of the initial conditions and forecast step varies significantly from [108] and [109] because we use the DMD of the ensemble forecast for our predictions. Now that the components of the GMM-DO filter are understood, we can discuss in further detail the GMM-DO filter and data assimilation with GMMs using the DO equations. This efficient, data-drive technique preserves non-Gaussian statistics and respects non-linear dynamics [109]. Throughout this discussion we utilize the notation shown in Table 3.1.

#### Forecast

The forecast that we use is based on DMD of the ensemble as discussed earlier. The approach allows for efficient extraction and forecasting which is advantageous for use where computing power and storage is limited. With an ensemble of  $N$  simulations used as training inputs,

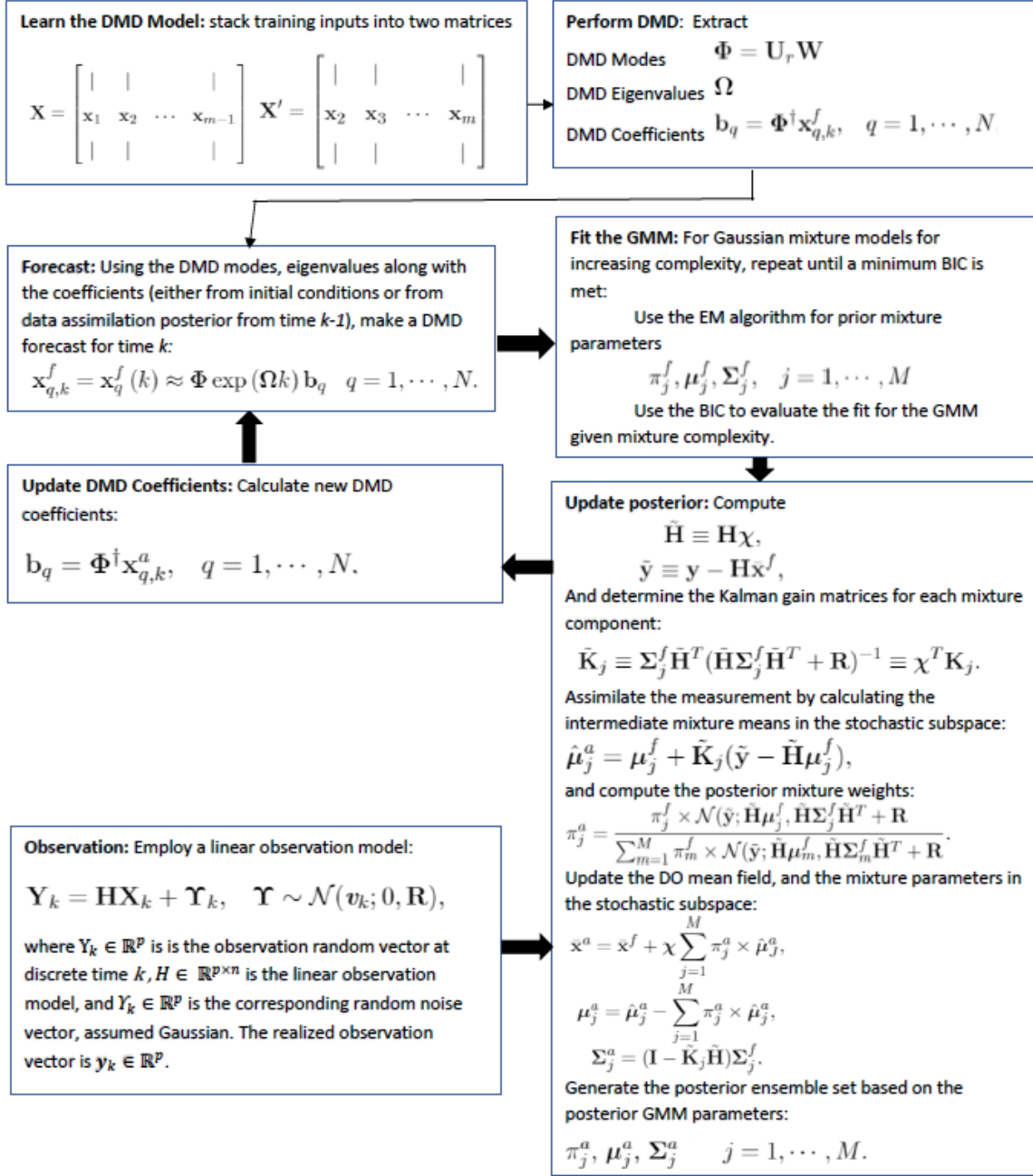


Figure 3-2: DMD with GMM-DO Filter flowchart.

Table 3.1: DMD with GMM-DO Filter Notation.

Descriptors		
$(\cdot)^f$		forecast or prior
$(\cdot)^a$		analysis or posterior
Scalars		
$i$	$\in \mathbb{N}$	stochastic subspace index
$j$	$\in \mathbb{N}$	mixture index
$k$	$\in \mathbb{N}$	discrete time index
$n$	$\in \mathbb{N}$	dimension of state vector
$p$	$\in \mathbb{N}$	dimension of observation vector
$q$	$\in \mathbb{N}$	realization index
$r$	$\in \mathbb{N}$	reduced DMD rank (number of DMD modes retained)
$s$	$\in \mathbb{N}$	dimension of stochastic subspace
$M$	$\in \mathbb{N}$	complexity of Gaussian Mixture Model
$N$	$\in \mathbb{N}$	number of Monte Carlo members
$\Psi_i$	$\in \mathbb{R}$	random variable describing probability density function for orthonormal mode $\tilde{\mathbf{x}}_i$
Vectors		
$\mathbf{x}$	$\in \mathbb{R}^n$	state (random) vector
$\tilde{\mathbf{x}}_i$	$\in \mathbb{R}^n$	modes describing an orthonormal basis for the stochastic subspace
$\bar{\mathbf{x}}$	$\in \mathbb{R}^n$	mean state vector
$\mathbf{x}_q$	$\in \mathbb{R}^n$	state realization
$\mathbf{Y}$	$\in \mathbb{R}^p$	observation (random) vector
$\mathbf{y}$	$\in \mathbb{R}^p$	observation realization
$\bar{\mathbf{x}}_j$	$\in \mathbb{R}^n$	mean vector of mixture j in state space
$\boldsymbol{\mu}_j$	$\in \mathbb{R}^s$	mean vector of mixture j in stochastic subspace
$\boldsymbol{\psi}_q$	$\in \mathbb{R}^s$	realization residing in stochastic subspace
$\boldsymbol{\Upsilon}$	$\in \mathbb{R}^p$	observation noise (random) vector
$\mathbf{b}_q$	$\in \mathbb{C}^r$	DMD coefficient vector of ensemble member $q$
Matrices		
$\mathbf{P}$	$\in \mathbb{R}^{n \times n}$	covariance matrix in state space
$\boldsymbol{\Sigma}_j$	$\in \mathbb{R}^{s \times s}$	covariance matrix of mixture j in stochastic subspace
$\mathbf{P}_j$	$\in \mathbb{R}^{n \times n}$	covariance matrix of mixture j in state space
$\mathbf{R}$	$\in \mathbb{R}^{p \times p}$	observation covariance matrix
$\mathbf{H}$	$\in \mathbb{R}^{p \times n}$	(linear) observation model
$\boldsymbol{\chi}$	$\in \mathbb{R}^{n \times s}$	matrix of orthonormal modes, $[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_s]$
$\{\boldsymbol{\psi}\}$	$\in \mathbb{R}^{s \times N}$	set of subspace ensemble realizations, $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \dots, \boldsymbol{\psi}_N\}$
$\boldsymbol{\Phi}$	$\in \mathbb{C}^{n \times r}$	set of DMD modes
$\boldsymbol{\Omega}$	$\in \mathbb{C}^{r \times r}$	diagonal matrix of discrete eigenvalues

projected DMD is applied to extract the POD modes  $\mathbf{U}_r$ , DMD modes  $\boldsymbol{\Phi}$ , and the DMD eigenvalues  $\lambda_i$  where  $i = 1, \dots, r$  and  $r$  is the reduced rank of the training inputs determined by the DMD algorithm. The DMD coefficients  $b_q$  for  $q$  ensemble members, are determined



based on initial conditions of each realization/ensemble member. Forecast at time  $k$  is then made by the following:

$$\mathbf{x}_{q,k}^f = \mathbf{x}_q^f(k) \approx \Phi \exp(\Omega k) \mathbf{b}_q \quad q = 1, \dots, N. \quad (3.69)$$

If no observations are made at time  $k$  then the DMD forecast  $\mathbf{x}_{q,k}^f$  is used by the vehicle as the forecast at time  $k$ . If, however, observations are available (either made by the platform or communicated to the platform) then GMM-DO is used for data assimilation. We start by placing the DMD forecasts into a matrix  $\mathbf{X}_k$ ;

$$\mathbf{X}_k = \begin{bmatrix} | & | & & | \\ \mathbf{x}_k^1 & \mathbf{x}_k^2 & \dots & \mathbf{x}_k^N \\ | & | & & | \end{bmatrix}. \quad (3.70)$$

We then determine the modes,  $\chi_k$ , and the subspace coefficients  $\{\psi_k\} = \{\psi_{1,k}, \dots, \psi_{N,k}\}$ :

$$\mathbf{x}_k^f = \bar{\mathbf{x}}_k^f + \chi_k^f \psi_k^f, \quad (3.71)$$

via a mean subtraction and compact SVD giving us:

$$\chi_k \Sigma_k^f \mathbf{V}_k^{fT} = \text{SVD}(\mathbf{x}_k^f - \bar{\mathbf{x}}_k^f), \quad (3.72)$$

and define the subspace coefficients as  $\Psi_k = \{\psi_k\} \equiv \Sigma_k^f \mathbf{V}_k^{fT}$ . Here,  $\chi_k \in \mathbb{R}^{n \times s}$  is the matrix of modes forming an orthonormal basis for the subspace at time  $k$ .

## Observation

For this thesis we employ a linear observation model:

$$\mathbf{Y}_k = \mathbf{H} \mathbf{X}_k + \Upsilon_k, \quad \Upsilon \sim \mathcal{N}(\mathbf{v}_k; 0, \mathbf{R}), \quad (3.73)$$

where  $\mathbf{Y}_k \in \mathbb{R}^p$  is the observation random vector at time  $k$ ,  $\mathbf{H} \in \mathbb{R}^{p \times n}$  is the linear observation model, and  $\Upsilon \in \mathbb{R}^p$  is the noise vector (assumed to be Gaussian). Note that the observation vector is denoted  $\mathbf{y}_k \in \mathbb{R}^p$  and the realized noise vector is denoted  $\mathbf{v}_k \in \mathbb{R}^p$ .

## Update

The update occurs at a discrete time instant and for simplicity we omit the time index  $k$  in the following. During the update, the modes are unchanged by the observations and thus no prior/forecast or posterior/analysis notation is used on the modes  $\chi$ . The goal is to update the mean state  $\bar{\mathbf{x}}^f$  and the ensemble members  $\{\boldsymbol{\psi}^f\} = \{\boldsymbol{\psi}_1^f, \dots, \boldsymbol{\psi}_N^f\}$  according to (3.73) and the observations  $\mathbf{y}$  to get the GMM-DO posterior estimate:

$$\mathbf{x}_q^a = \bar{\mathbf{x}}^a + \chi \boldsymbol{\psi}_q^a, \quad q = \{1, \dots, N\}. \quad (3.74)$$

*GMM representations of prior ensemble set:* at discrete time when new set of observations or measurements are obtained, the EM algorithm and BIC allow us to conclude the GMM that best expresses the ensemble set within the stochastic subspace which is significantly less expensive than doing this in the full state space. The parameters of the GMM are denoted:

$$\pi_j^f, \boldsymbol{\mu}_j^f, \boldsymbol{\Sigma}_j^f, \quad j = 1, \dots, M, \quad (3.75)$$

where  $\pi_j^f \in [0, 1]$  are the component weights,  $\boldsymbol{\mu}_j^f \in \mathbb{R}^s$  is the mean vector of mixture component  $j$  in stochastic subspace, and  $\boldsymbol{\Sigma}_j^f \in \mathbb{R}^{s \times s}$  is the covariance matrix of mixture component  $j$  in stochastic subspace.

The best mixture complexity is decided by the BIC successively fitting GMMs of increasing complexity using the EM algorithm until a minimum BIC is determined. The end result is a GMM that is best fit to the ensemble set in the stochastic subspace. The resulting prior pdf can be written as follows:

$$p_{\boldsymbol{\Psi}^f}(\boldsymbol{\psi}^f) = \sum_{j=1}^M \pi_j^f \times \mathcal{N}(\boldsymbol{\psi}^f; \boldsymbol{\mu}_j^f, \boldsymbol{\Sigma}_j^f). \quad (3.76)$$

A very important property of the GMM-DO filter is that the GMM can be expanded into the state space by:

$$\bar{\mathbf{x}}_j^f = \bar{\mathbf{x}}^f + \chi \boldsymbol{\mu}_j^f \quad (3.77)$$

and

$$\mathbf{P}_j^f = \chi \boldsymbol{\Sigma}_j^f \chi^T. \quad (3.78)$$

The mixture weights remain unchanged when expanding to the state space. The GMM prior distribution in the state space then takes on the form:

$$p_{\mathbf{x}^f}(\mathbf{x}^f) = \sum_{j=1}^M \pi \times \mathcal{N}(\mathbf{x}^f; \bar{\mathbf{x}}_j^f, \mathbf{P}_j^f). \quad (3.79)$$

The great thing about GMM-DO is that the same distribution would have been obtained if we had performed the fitting to the state space (but the fitting in the subspace is done at a fraction of the cost).

*Bayesian Update:* Since the dimension of the subspace is much less than the state space ( $s \ll n$ ), for realistic problems performing a GMM-DO filter update step is only computationally feasible in the subspace. The following is the update step that determines the posterior distribution in the stochastic subspace  $p_{\Psi^a}(\psi^a)$  but would be equivalent to evolving the distribution in the full state space because the modes  $\chi$  are unchanged (proof is shown in [108]).

Specifically we show the updates of the mean  $\bar{\mathbf{x}}^f$  and parameters  $\pi_j^f$ ,  $\boldsymbol{\mu}_j^f$ , and  $\boldsymbol{\Sigma}_j^f$ :

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^f + \chi \sum_{j=1}^M \pi_j^a \times \hat{\boldsymbol{\mu}}_j^a, \quad (3.80)$$

$$= \bar{\mathbf{x}}^f + \chi \sum_{j=1}^M \pi_j^a \times [\boldsymbol{\mu}_j^f + \tilde{\mathbf{K}}_j(\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\boldsymbol{\mu}_j^f)], \quad (3.81)$$

$$\pi_j^a = \frac{\pi_j^f \times \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{H}}\boldsymbol{\mu}_j^f, \tilde{\mathbf{H}}\boldsymbol{\Sigma}_j^f\tilde{\mathbf{H}}^T + \mathbf{R})}{\sum_{m=1}^M \pi_m^f \times \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{H}}\boldsymbol{\mu}_m^f, \tilde{\mathbf{H}}\boldsymbol{\Sigma}_m^f\tilde{\mathbf{H}}^T + \mathbf{R})}, \quad (3.82)$$

$$\boldsymbol{\mu}_j^a = \hat{\boldsymbol{\mu}}_j^a - \sum_{j=1}^M \pi_j^a \times \hat{\boldsymbol{\mu}}_j^a, \quad (3.83)$$

$$\boldsymbol{\Sigma}_j^a = (\mathbf{I} - \tilde{\mathbf{K}}_j\tilde{\mathbf{H}})\boldsymbol{\Sigma}_j^f, \quad (3.84)$$

where the following definitions apply and  $\mathbf{I}$  is the identity matrix of appropriate dimensions:

$$\tilde{\mathbf{H}} \equiv \mathbf{H}\chi, \quad (3.85)$$

$$\tilde{\mathbf{y}} \equiv \mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^f, \quad (3.86)$$

$$\hat{\boldsymbol{\mu}}_j^a = \boldsymbol{\mu}_j^f + \tilde{\mathbf{K}}_j(\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\boldsymbol{\mu}_j^f), \quad (3.87)$$

$$\tilde{\mathbf{K}}_j \equiv \boldsymbol{\Sigma}_j^f \tilde{\mathbf{H}}^T (\tilde{\mathbf{H}} \boldsymbol{\Sigma}_j^f \tilde{\mathbf{H}}^T + \mathbf{R})^{-1} \equiv \boldsymbol{\chi}^T \mathbf{K}_j. \quad (3.88)$$

*Generating the posterior ensemble set:* We complete the update by utilizing the same idea as in ESSE Scheme A [66] by generating a posterior ensemble set in the stochastic subspace  $\{\boldsymbol{\psi}^a\} = \{\boldsymbol{\psi}_1^a, \dots, \boldsymbol{\psi}_N^a\}$  in accordance with the posterior GMM,  $p_{\boldsymbol{\Psi}^a}(\boldsymbol{\psi}^a)$ , that has the parameters  $\pi_j^a, \boldsymbol{\mu}_j^a, \boldsymbol{\Sigma}_j^a$  for  $j = 1, \dots, M$ .

Now we have arrived at the posterior DO representation for the state vector based on Bayesian data assimilation of the observation  $\mathbf{y}$  at time  $k$  as:

$$\mathbf{x}_{q,k}^a = \bar{\mathbf{x}}_k^a + \boldsymbol{\chi}_k \boldsymbol{\psi}_{q,k}^a, \quad q = \{1, \dots, N\}. \quad (3.89)$$

### DMD Coefficients Updates

Now that the update is complete and a posterior ensemble set is obtained, we can then utilize the DMD modes,  $\boldsymbol{\Phi}$  and the new ensemble set to obtain new DMD coefficients,  $\mathbf{b}_q$ . If time  $k$  is set to 0, the the coefficients are:

$$\mathbf{b}_q = \boldsymbol{\Phi}^\dagger \mathbf{x}_{q,k}^a, \quad q = 1, \dots, N, \quad (3.90)$$

and the forecast at some time  $t$  in the future is:

$$\mathbf{x}_{q,t}^f = \mathbf{x}_q^f(t) \approx \boldsymbol{\Phi} \exp(\boldsymbol{\Omega}t) \mathbf{b}_q \quad q = 1, \dots, N. \quad (3.91)$$

## 3.5 Summary

To conclude, we reviewed some DA schemes focusing on EnKF and GMM-DO where the DA occurs in the stochastic subspace for efficiency (like in the ESSE technique). Additionally we presented the novel technique of combining DMD modeling with the GMM-DO data assimilation scheme. We have described the DMD with GMM-DO filter as an efficient, data-driven assimilation scheme that respects nonlinear dynamics and captures non-Gaussian statistics. In the following chapters, we apply the DMD with GMM-DO filter and DMD with EnKF to test cases for comparison.

## Chapter 4

# DMD with DA Test Cases and Applications

In this chapter we present two test cases and applications of our novel Bayesian DA where the prior is a DMD forecast. The first test case is a flow behind an island/cylinder test case with a Reynolds's number (Re) of approximately 200. Below we describe in some detail how we generate this test case and our results from utilizing DMD, DMD with EnKF and DMD with GMM-DO.

The second, more realistic, test case included in this chapter, comes from high fidelity simulation snapshots from the POSYDON-POINT experiment [65]. These simulations consist of temperature, salinity,  $u$ -velocity, and  $V$ -velocity hourly simulations for 96 hour period in the Middle Atlantic - New York Bight region. from 23 August to 27 August 2018. The ensemble set is a total of 300 members.

We start with the flow behind an island/cylinder vorticity test case in the following section.

### 4.1 Flow Behind an Island/Cylinder Test Case

In this example, we demonstrate DMD and DMD with DA on simulations representing a time series of fluid vorticity fields for flow behind a circular island/cylinder at a Reynolds number  $Re = 200$ . The Reynolds number is a unit-less quantification of the ratio of inertial viscous forces that is defined as  $Re = \frac{DU}{\nu}$ , where  $D$  is the diameter of the island/cylinder,  $U$  is the free-stream velocity (in this example in the positive horizontal direction), and  $\nu$

is the kinematic fluid viscosity. We chose a Reynolds number above the critical Reynolds number  $\text{Re}_{\text{crit}} \approx 47$ , because at this point the flow undergoes a Hopf bifurcation resulting in laminar vortex shedding [124]. This cycle is stable and is representative of three-dimensional flow [92].

#### 4.1.1 Description of Test Case

This two-dimensional Navier-Stokes equation solver was adapted from course 2.29 code from the Massachusetts Institute of Technology and the following description uses the course notes of 2.29 [63] as well as [20] for description. Note that several works use the same equation solver (albeit with different parameters) for test cases [76, 78, 114],

### 3-Dimensional Navier Stokes Equation

Before discussing the 2-dimensional flow that we generated with the following finite volume framework, we first state the governing equations in the general 3-dimensional framework so we can discuss how 3-dimensional flow can be simplified to 2-dimensional flow and simulated. The general 3-dimensional incompressible Navier-Stokes (N-S) equations are [20]:

$$\nabla \cdot \mathbf{u} = 0, \quad (4.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} - 2\mathbf{\Omega} \times \mathbf{u} + \mathbf{F}_u(\mathbf{x}, t), \quad (4.2)$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \kappa_\phi \nabla^2 \phi + F_\phi(\phi, \mathbf{x}, t), \quad (4.3)$$

where  $\mathbf{u}$  is fluid velocity field,  $p$  is the pressure field,  $\nu$  is the kinematic viscosity,  $\rho$  is the density,  $\nabla$  is the gradient operator,  $\nabla^2$  is the Laplace operator,  $\mathbf{g}$  is a constant gravitational acceleration,  $\mathbf{\Omega}(\mathbf{x}, t)$  is a given angular velocity vector field under the earth rotation framework,  $\mathbf{F}_u$  is a given source term representing other known exterior forcing,  $\phi$  is an arbitrary tracer (a scalar field transported by advection and diffusion,  $\kappa_\phi$  is a given diffusivity constant of  $\phi$ , and  $F_\phi$  is a given source term which can also depend on  $\phi$ ).

### N-S Equations with Boussinesq Approximation

Generally (4.1) and (4.2) form a closed system and (4.3) is coupled to them in a one-way manner. Yet there is a frequent naturally occurring situation where a tracer also goes into

the momentum equation (4.2) and thus fully couples (4.1), (4.2), and (4.3). This case, like natural convection in the atmosphere and bottom gravity current in the ocean, is density driven flow.

In these density driven flows, the dominant driving factor is buoyancy caused by the non-uniformly distributed fluid density along with gravity. The density variation could be mainly due to factors like temperature or salinity or both, but normally the pressure effect is minimal. The density fluctuation is typically small compared to the absolute value of density itself (e.g.  $< 1\%$  in the ocean). In these cases, density variation is still very important and cannot be ignored. On the other hand, the compressibility effect, namely pressure wave, is still negligible. Hence, we want to approximate the flow in such a way that it can still be treated as incompressible while still accounting for the effects of gravity exerted on fluid with different densities. The Boussinesq approximation [9] does this.

To describe Boussinesq approximation we start with rewriting the momentum equation (4.2) for incompressible flow as [20]:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \rho \nu \nabla^2 \mathbf{u} + \rho \mathbf{g} - 2\rho \boldsymbol{\Omega} \times \mathbf{u} + \rho \mathbf{F}_u(\mathbf{x}, t). \quad (4.4)$$

With the density  $\rho$  having only small variation, we rewrite it as:

$$\rho = \rho_0 + \rho' \quad (4.5)$$

where the background part is  $\rho_0$  and the perturbation is  $\rho'$ . We use the same idea to decompose pressure  $p$  into an inactive part  $p_0$  that is due to  $\rho_0$  and the perturbation  $p'$  caused by  $\rho'$ :

$$p = p_0 + p' \quad (4.6)$$

where

$$p_0(\mathbf{x}) = p_{0,\text{ref}} + \rho_0 \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_{0,\text{ref}}) \quad (4.7)$$

and  $\mathbf{x}_{0,\text{ref}}$  is an arbitrary reference point and  $p_{0,\text{ref}}$  is a reference pressure.

With equations (4.5), (4.6), and (4.7), the pressure and gravity terms in (4.4) can be rewritten as:

$$-\nabla p + \rho \mathbf{g} = -\nabla(p_{0,\text{ref}} + \rho_0 \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_{0,\text{ref}}) + p') + (\rho_0 + \rho') \mathbf{g} = -\nabla p' + \rho' \mathbf{g}. \quad (4.8)$$

Now we can substitute (4.5) and (4.8) into the momentum equation (4.4) and divide by  $\rho_0$  to get:

$$\left(1 + \frac{\rho'}{\rho_0}\right) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\frac{1}{\rho_0} \nabla p' + \frac{\rho'}{\rho_0} \mathbf{g} + \left(1 + \frac{\rho'}{\rho_0}\right) (\nu \nabla^2 \mathbf{u} - 2\boldsymbol{\Omega} \times \mathbf{u} + \mathbf{F}_u(\mathbf{x}, t)). \quad (4.9)$$

Since the Boussinesq approximation tells us that if  $|\rho'| \ll \rho_0$ , the effects of  $\rho'$  are negligible except in the gravity term. This allows for (4.9) to be approximated well by:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0} \nabla p' + \nu \nabla^2 \mathbf{u} + \frac{\rho'}{\rho_0} \mathbf{g} - 2\boldsymbol{\Omega} \times \mathbf{u} + \mathbf{F}_u(\mathbf{x}, t). \quad (4.10)$$

In the following, because  $\rho_0$  is a constant, for convenience we will define a reduced pressure,  $P'$ , and reduced density,  $\tilde{\rho}'$ , as:

$$P' = \frac{p'}{\rho_0}, \quad \tilde{\rho}' = \frac{\rho' g}{\rho_0}. \quad (4.11)$$

## 2-Dimensional Flow Idealization and Nondimensionalization

Now that we have discussed the 3-dimensional N-S equation and Boussinesq approximation, now we can show the 2-dimensional case. The 2-dimensional flow in the horizontal plane is governed by [20]:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (4.12)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2} + \nu \frac{\partial^2 u}{\partial y^2} + f(x, y, t)v + F_u(x, y, t), \quad (4.13)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \nu \frac{\partial^2 v}{\partial x^2} + \nu \frac{\partial^2 v}{\partial y^2} + f(x, y, t)u + F_v(x, y, t), \quad (4.14)$$

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} = \kappa \frac{\partial^2 \rho}{\partial x^2} + \kappa \frac{\partial^2 \rho}{\partial y^2} + F_\rho(\rho, x, y, t). \quad (4.15)$$

The 2-dimensional flow in the vertical plane is governed by [20]:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0, \quad (4.16)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = -\frac{\partial P'}{\partial x} + \nu_{xy} \frac{\partial^2 u}{\partial x^2} + \nu_z \frac{\partial^2 u}{\partial z^2} + F_u(x, z, t), \quad (4.17)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{\partial P'}{\partial z} + \nu_{xy} \frac{\partial^2 w}{\partial x^2} + \nu_z \frac{\partial^2 w}{\partial z^2} - \frac{\rho' g}{\rho_0} + F_w(x, z, t), \quad (4.18)$$



$$\frac{\partial \rho'}{\partial t} + u \frac{\partial \rho'}{\partial x} + w \frac{\partial \rho'}{\partial z} = \kappa_{xy} \frac{\partial^2 \rho'}{\partial x^2} + \kappa_z \frac{\partial^2 \rho'}{\partial z^2} + F_\rho(\rho', x, z, t). \quad (4.19)$$

Finally by removing the hydrostatic pressure and fixing the surface  $z = b$  we get:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0, \quad (4.20)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = -\frac{\partial P'}{\partial x} - \frac{g}{\rho_0} \int_z^b \frac{\partial \rho'}{\partial x} dz + \nu_{xy} \frac{\partial^2 u}{\partial x^2} + \nu_z \frac{\partial^2 u}{\partial z^2} + F_u(x, z, t), \quad (4.21)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{\partial P'}{\partial z} + \nu_{xy} \frac{\partial^2 w}{\partial x^2} + \nu_z \frac{\partial^2 w}{\partial z^2} + F_w(x, z, t), \quad (4.22)$$

$$\frac{\partial \rho'}{\partial t} + u \frac{\partial \rho'}{\partial x} + w \frac{\partial \rho'}{\partial z} = \kappa_{xy} \frac{\partial^2 \rho'}{\partial x^2} + \kappa_z \frac{\partial^2 \rho'}{\partial z^2} + F_\rho(\rho', x, z, t). \quad (4.23)$$

### Solving the N-S equations using our MSEAS Finite Volume Framework

To generate our simulations, we use the framework presented in [121] and [63]. We start by solving the following equations [63]:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (4.24)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \nu_1 \frac{\partial^2 u}{\partial x^2} + \nu_2 \frac{\partial^2 u}{\partial y^2} + f(x, y, t)v + F_u(x, y, t), \quad (4.25)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \nu_1 \frac{\partial^2 v}{\partial x^2} + \nu_2 \frac{\partial^2 v}{\partial y^2} - \rho g - f(x, y, t)u + F_v(x, y, t), \quad (4.26)$$

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} = \kappa_1 \frac{\partial^2 \rho}{\partial x^2} + \kappa_2 \frac{\partial^2 \rho}{\partial y^2} + F_\rho(\rho, x, y, t). \quad (4.27)$$

These equations are solved using a Total Variation Diminishing (TVD) scheme with a monotonized central (MC) symmetric flux limiter [122]. The TVD scheme is effectively a combination of a central difference scheme (CDS) and an upwind (UW) scheme. The TVD scheme can be for a variable  $\eta$  as:

$$F\left(\eta_{i-\frac{1}{2}}\right) = u_{i-\frac{1}{2}} \frac{\eta_i + \eta_{i-1}}{2} - \left|u_{i-\frac{1}{2}}\right| \left(\frac{\eta_i - \eta_{i-1}}{2}\right) \left[1 - \left(1 - \left|u_{i-\frac{1}{2}} \frac{\Delta t}{\Delta x}\right|\right) \Gamma(r_{i-\frac{1}{2}})\right], \quad (4.28)$$

where the MC slope limiter  $\Gamma(r)$  is:

$$\Gamma(r) = \max \left\{ 0, \min \left[ \min \left( \frac{1+r}{2}, 2 \right), 2r \right] \right\}, \quad (4.29)$$

and the variable  $r$  is defined as:

$$r_{i-\frac{1}{2}} = \frac{\left[ \frac{1}{2} \left( u_{i-\frac{1}{2}} + \left| u_{i-\frac{1}{2}} \right| \right) (\eta_{i-1} - \eta_{i-2}) + \frac{1}{2} \left( u_{i-\frac{1}{2}} - \left| u_{i-\frac{1}{2}} \right| \right) (\eta_{i+1} - \eta_i) \right]}{u_{i-\frac{1}{2}} (\eta_i - \eta_{i-1})}, \quad (4.30)$$

where  $u_{i-\frac{1}{2}}$  is used without interpolation for the density advection (due to grid staggering) while the second-order linear interpolation is used for the non-linear  $u$  and  $v$  advection. For more information on TVD schemes refer to [67].

## Parameters and Setup

The domain  $3 \text{ m} \times 20 \text{ m}$ . In this case the grid contains  $20 \times 170$  points. A time step of  $\Delta t = 0.01 \text{ s}$ , was used because it satisfied the Courant–Friedrichs–Lewy condition [17]. We use a kinematic viscosity of  $0.01 \text{ m}^2/\text{s}$  and a diffusivity of 0. We used a cylinder of diameter  $1 \text{ m}$ . The general setup of the domain can be seen in Fig. 4-1.

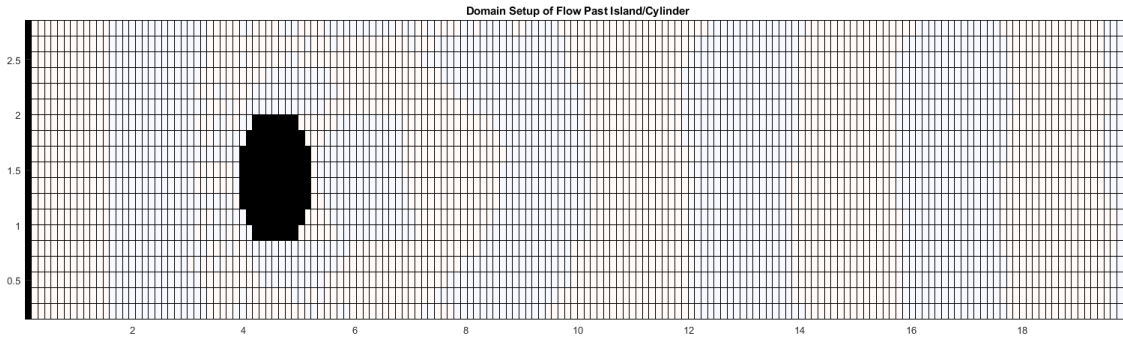


Figure 4-1: General setup of domain for flow behind an island/cylinder.

To generate our numerical simulation we utilized the idea of the identical twin experiment [55]. A numerical model simulation during a certain time interval is chosen to be the 'true' or control run. In this case we generated a 'true' snapshots of the flow with a free stream velocity of  $2 \text{ m/s}$ . A subsampled dataset is then extracted from this simulation as our observations. Additionally, to generate the ensemble of inputs for testing our learning with DA schemes, we generated an ensemble with a mean free stream velocity of  $2 \text{ m/s}$  with perturbations added where the ensemble free stream velocity has a standard deviation

of  $0.2 \text{ m/s}$ . The 'true' field has a Reynolds number of 200 while some of the perturbed ensembles have Reynolds numbers slightly lower and some slightly higher based on the free stream velocity.

Although with the above framework we solve for  $u$ ,  $v$ ,  $\rho$ , and  $P$  at each grid-point, in the following example we have used  $u$  and  $v$  to calculate the vorticity at each point and use this vorticity as our training inputs and our observations, as appropriate.

### 4.1.2 Simulated Flow Snapshots and Measurements

A snapshot of the vorticity field is shown in top plot of Fig. 4-2 to exemplify a sample vorticity field (top plot) and two sensors (bottom plot) for flow behind an island/cylinder at  $\text{Re} = 200$  for demonstrative purposes. The bottom plot of Fig. 4-2 shows the vorticity at sensor one (labeled  $s_1$ ) vs. time in a blue solid line and the vorticity at sensor 2 (labeled  $s_2$ ) vs. time in an orange dotted line.

After the simulations converge to steady-state vortex shedding, snapshots are collected at regular intervals of  $20\Delta t$  or  $0.2 \text{ s}$ . These snapshots are then arranged into matrices where each column represents a snapshot in time so that they can be used with DMD and DMD with DA algorithms. The columns are stacked as in equations (2.36) and (2.37) and DMD is performed on this matrices. The DMD coefficients of each ensemble member are then determined based on the dominant DMD modes of the ensemble. For the following, we utilize 25 snapshots for training (5s worth of training snapshots of every  $0.2 \text{ s}$ ) and we forecast each  $0.2 \text{ s}$  for  $25 \text{ s}$  (125 snapshots).

### 4.1.3 Ensemble POD Modes

In Fig. 4-3, we show the mean vorticity field, the first 50 singular values, and the first eight most dominant POD modes of the ensemble. Although the singular value decomposition (SVD) of the ensemble is a separation of variables resulting from spatial-temporal decomposition, it is still able to approximate the dominant structures required for the traveling wave solution to the island/cylinder flow. It is possible to apply POD to a variety of variables such as velocity, temperature, salinity, etc. In this case we are focusing on the vorticity.

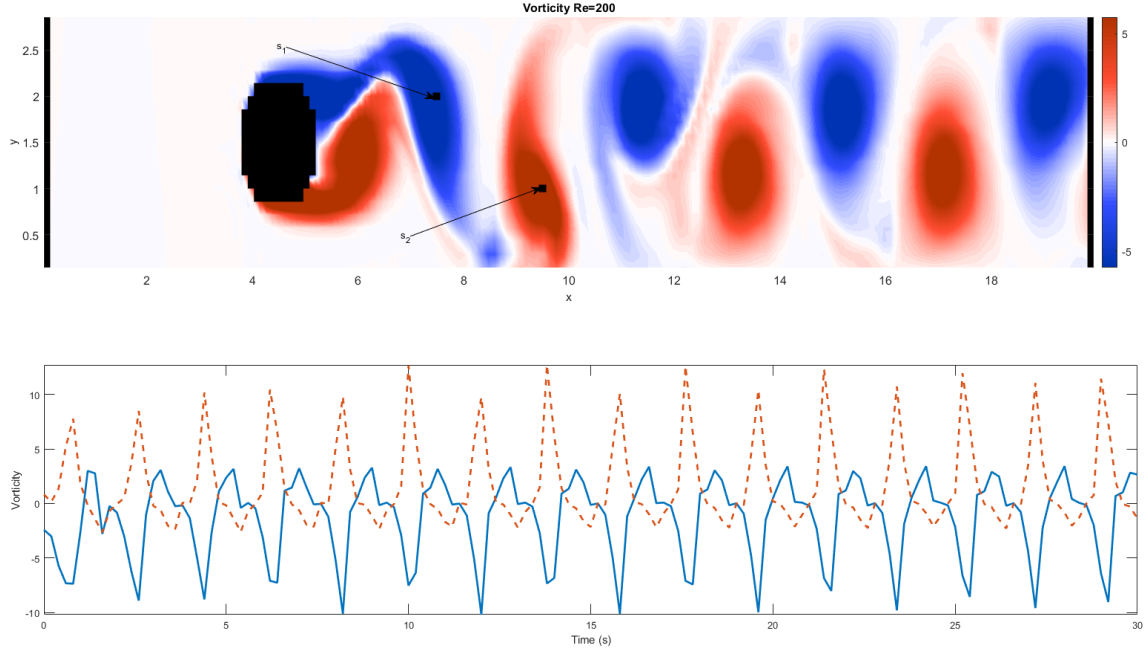


Figure 4-2: Example of vorticity field (top plot) and two sensors (bottom plot) for flow behind an island/cylinder at  $Re = 200$ . Bottom plot shows the vorticity at sensor one (labeled  $s_1$ ) vs. time in a blue solid line and the vorticity at sensor 2 (labeled  $s_2$ ) vs. time in an orange dotted line.

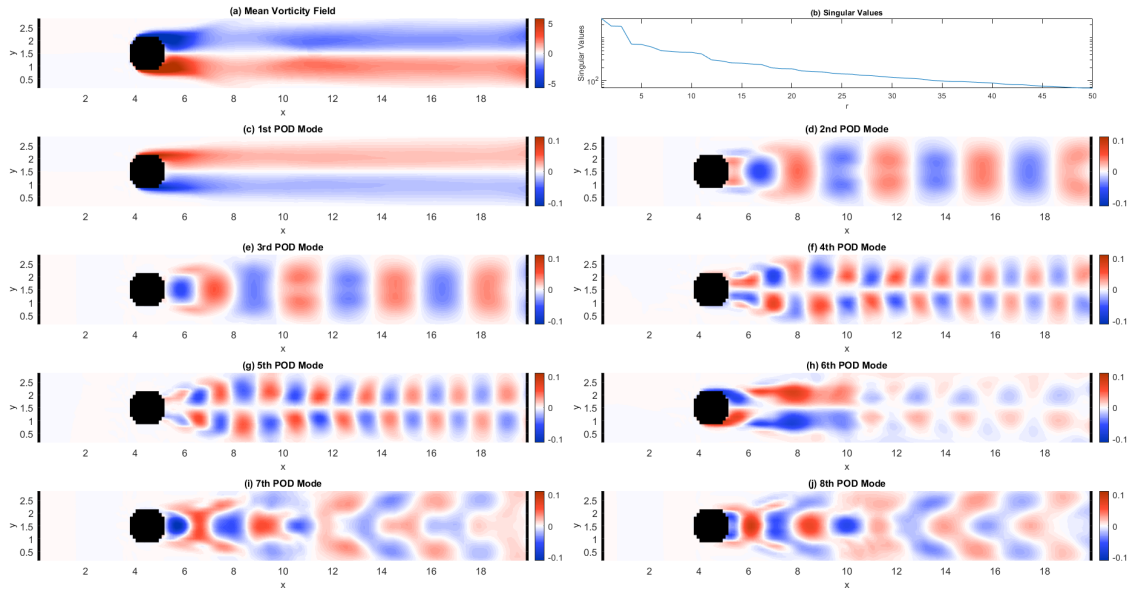


Figure 4-3: Mean vorticity field (a) and POD singular values (b) for flow behind an island/cylinder with  $Re = 200$ . The first 6 POD modes are shown in (c)-(j).

#### 4.1.4 DMD with DA Results for Flow Behind an Island/Cylinder

To apply DMD to the ensemble for flow behind an island/cylinder it requires the same snapshot information that we used for POD and is a purely data-driven method. In the following we show that DMD can extract dominant modes that can be used for prediction. These DMD modes capture the dominant spatial-temporal structures and the underlying dynamics. The fact that there are perturbations in the ensembles initial conditions though, often leads to the DMD solutions relying too heavily on the initial conditions and thus the DMD solutions (and ensemble mean DMD solution) drifting from the truth. This is where the DA helps.

Fig. 4-4 exemplifies the performance of the DMD of the ensemble method at extracting underlying dynamics from an ensemble. If we were to perform DMD on an individual ensemble member it is extremely unlikely we would get similar dominant DMD modes. On the other hand, the DMD of the ensemble extracted very similar dominant DMD modes (the real part of which are shown in the figure).

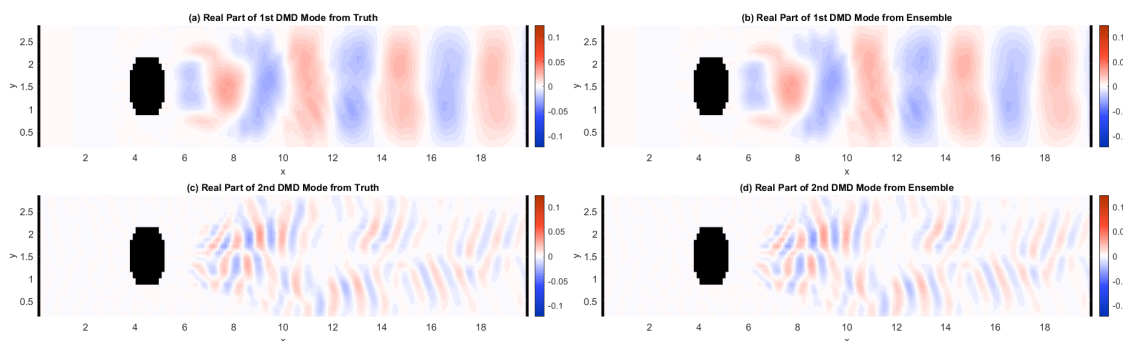


Figure 4-4: Real part of the first two Dominant DMD modes. (a) and (c) are extracted from the 'truth' where as (b) and (d) are extracted from the ensemble.

#### 4.1.5 DMD Forecasts

At this point we utilize our DMD of the ensemble method to predict the future ensemble set as we have discussed. Repeating that the test case has 30 ensemble members, 25 training snapshots, and predictions of the ensemble were made 125 snapshots past the last training snapshot. For visual comparison we calculated the ensemble mean forecast and compared it to the 'truth' at the specified time in the future. The mean error field is calculated and displayed as well as the pattern correlation coefficient (PCC). The PCC is a time mean

subtracted pattern coefficient [55]. In general terms, a PCC of 1 is a perfect correlation, 0 no correlation, and -1 a perfect anti-correlation. Fig. 4-5, Fig. 4-6, and Fig. 4-7 show the true field, DMD error field, and PCC values for 0.2 s, 5 s, and 20 s forecast respectively. As we can see the DMD forecast starts off with a relatively high PCC and low error fields at short forecast times, but with 5 s the error fields are much larger, and by 20s the DMD forecast is very poor. We can see that due to the somewhat cyclical nature of the wake behind the island/cylinder, persistence does a poor job but gets lucky as the pattern repeats itself.

These results are similar to what we see in ocean modeling for DMD [41]. This is why utilizing cost effective data assimilation schemes prove very beneficial for use with DMD onboard autonomous platforms.

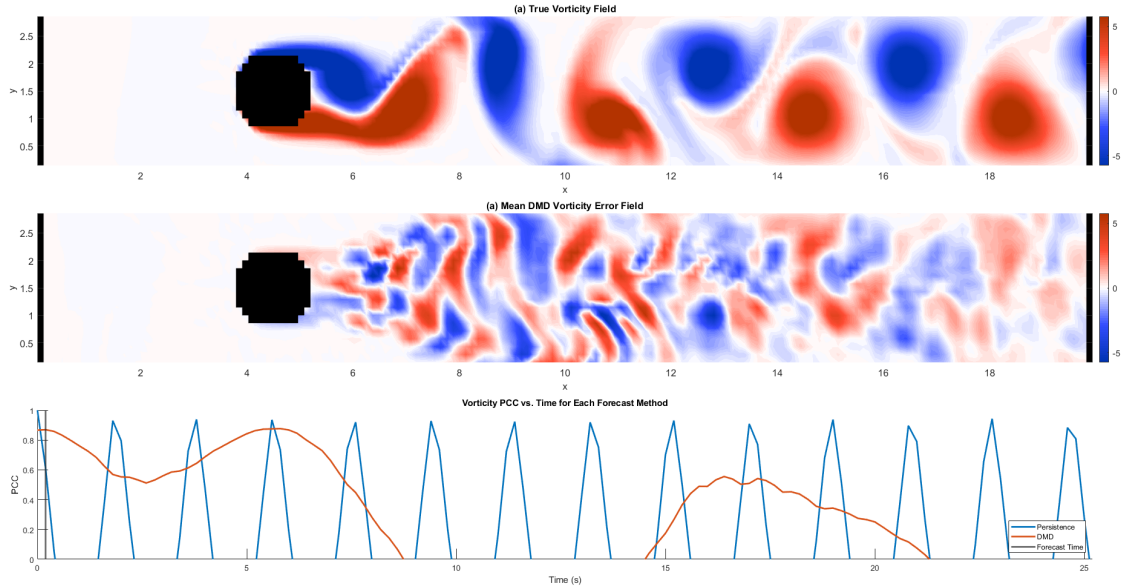


Figure 4-5: True vorticity field 0.2 s past the last training snapshot (top plot), DMD error field at 0.2 s past the last training snapshot, and the pattern correlation coefficients (PCCs) for persistence and DMD.

#### 4.1.6 DMD with DA Results

As we have demonstrated and shown above, the DMD forecasts are not reliable after a relatively short period of time. It is not necessarily that the underlying dynamics learned by DMD are wrong (as we have shown the dominant DMD modes are extremely similar to those of the truth). When we assimilate data and use this data to calculate new coefficients

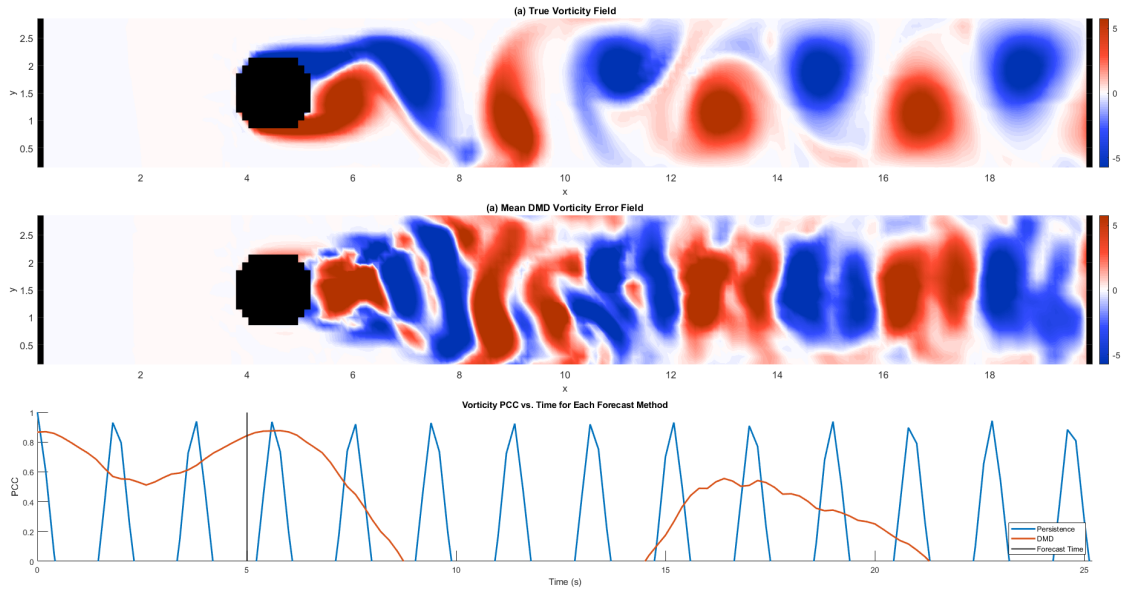


Figure 4-6: Same as Fig. 4-5 except 5 s past last training snapshot.

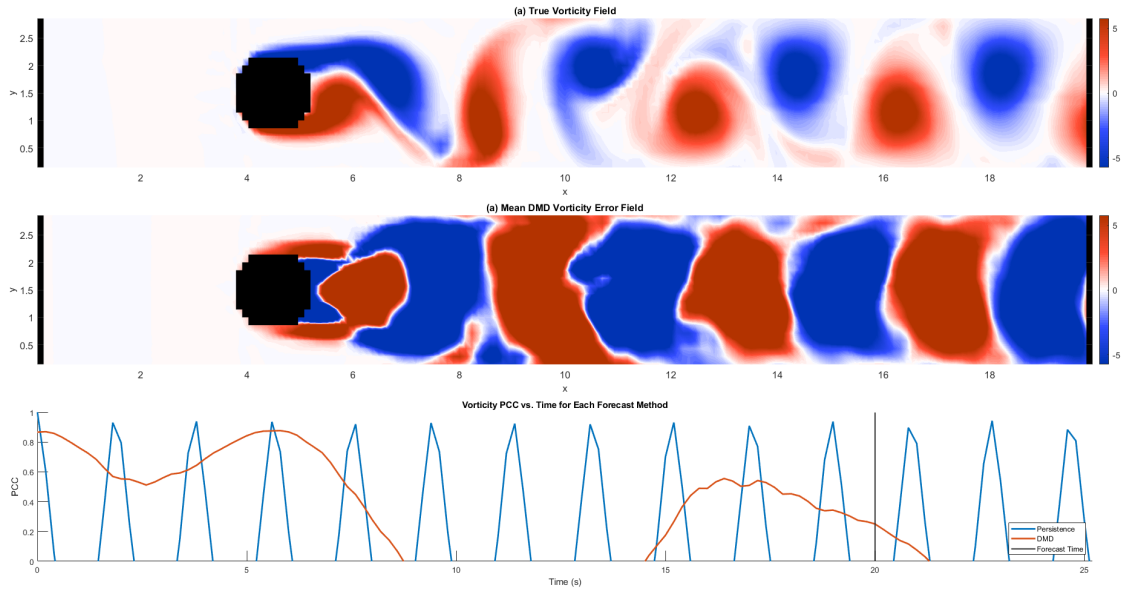


Figure 4-7: Same as Fig. 4-5 except 20 s past last training snapshot.

to make predictions we can drastically improve our field predictions.

In the following example we have observed the vorticity at 24 grid-points at every 0.2 s. These observations are drawn at random from the domain (but do not necessarily have to if the observations were made only by the vehicle itself).

Figure 4-8, Fig. 4-9, and Fig. 4-10 show the true field, DMD with EnKF error field, DMD with GMM-DO error field, and PCC values for 0.2 s, 5 s, and 20 s forecast respectively. Here we can see that in general the EnKF performs slightly better than DMD forecasts in most instances. On the other hand, the GMM-DO filter performs much better than the EnKF for the duration of our test. The reasons for this are likely that the ensembles do not have a Gaussian distribution and that a GMM better represents the true vorticity ensemble distribution.

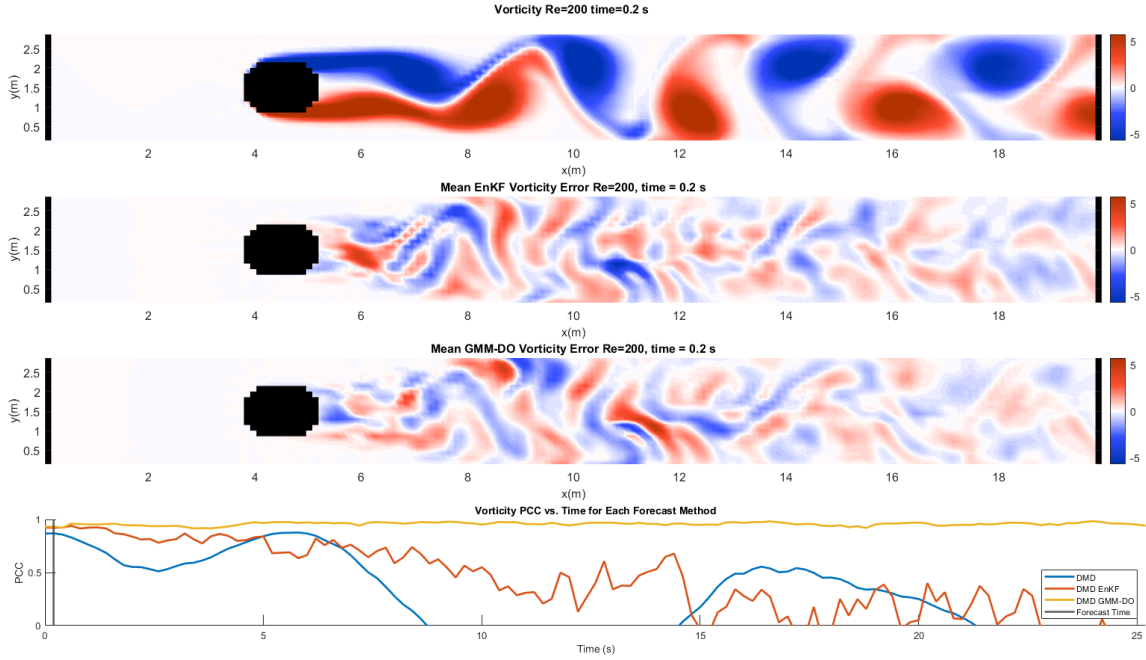


Figure 4-8: True velocity field 0.2 s past the last training snapshot (top plot), DMD with EnKF error field at 0.2 s past the last training snapshot (second from top), DMD with EnKF error field at 0.2 s past the last training snapshot (second from bottom), and the pattern correlation coefficients (PCCs) for DMD, DMD with EnKF, and DMD with GMM-DO.



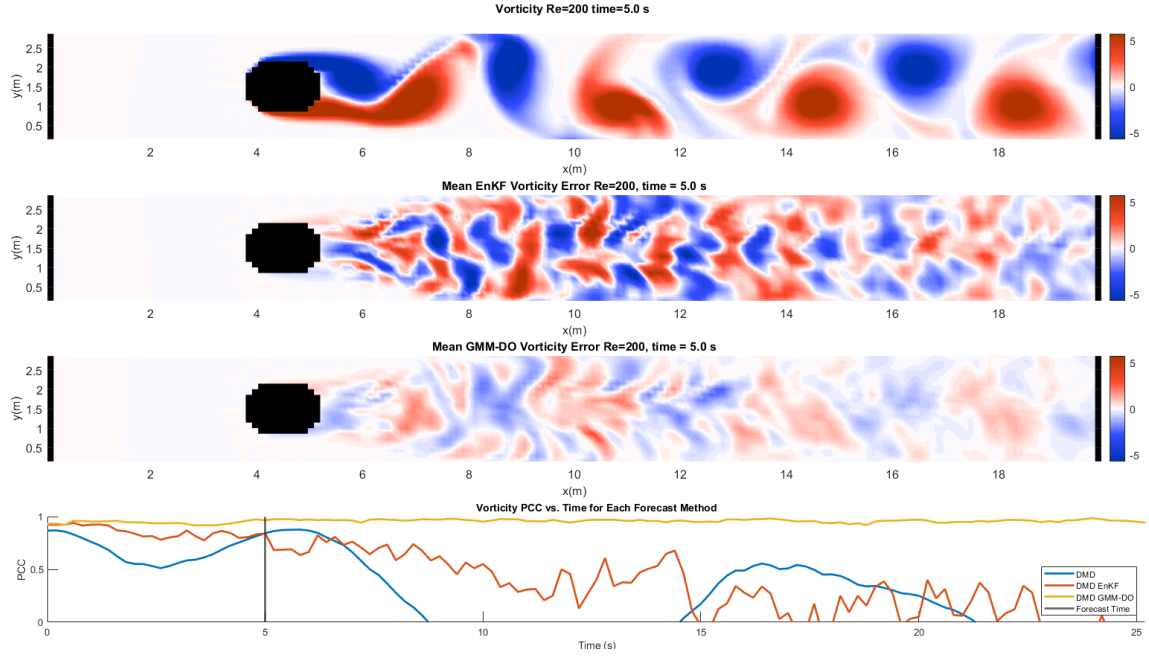


Figure 4-9: Same as Fig. 4-8 but at 5 s past the last training snapshot.

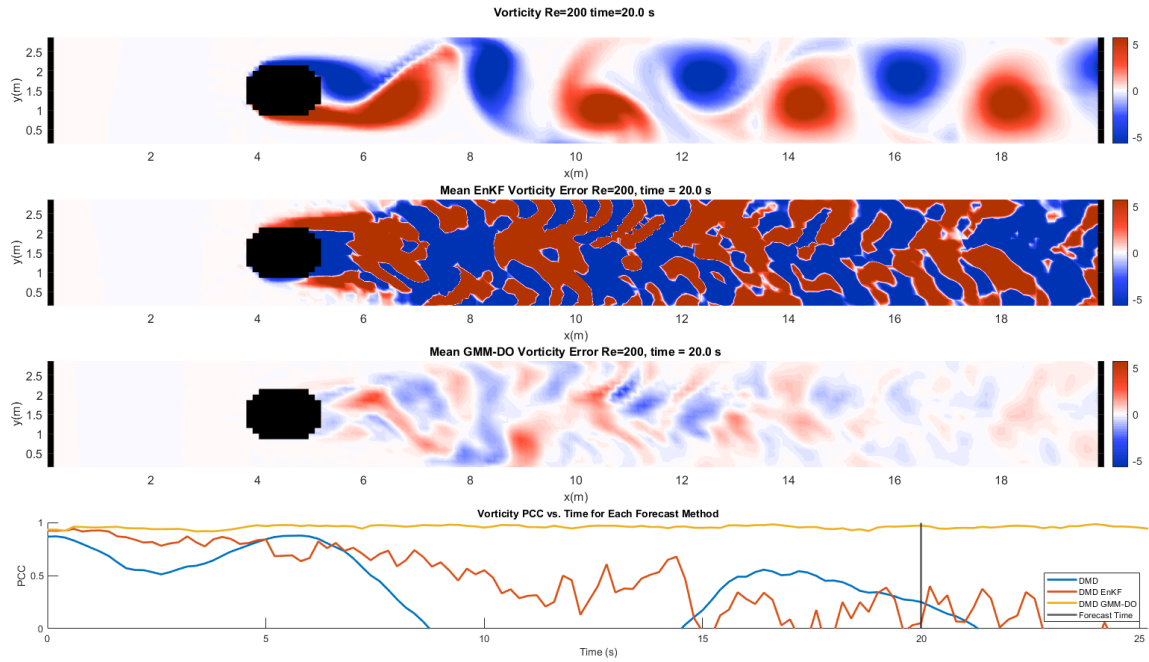


Figure 4-10: Same as Fig. 4-8 but at 20 s past the last training snapshot.

## 4.2 Realistic Ocean Simulation Test Case

### 4.2.1 Description of POSYDON-POINT Test Case

The BBN POSYDON-POINT simulations [65] were developed using the probabilistic MSEAS Primitive Equation (PE) modeling system [89, 38, 37] to provide ocean field and uncertainty forecasts at a 3 km grid resolution creating a 300-member ensemble tuned for region specific uncertainty modeling using the ESSE methodology [58]. The ocean forecasts are initialized from HYCOM [16] and down-scaled to higher resolution. These ocean simulations are forced by atmospheric flux fields forecast by the global forecast system (GFS) 0.25° model from the National Centers for Environmental Prediction (NCEP) [32] and tidal forcing from TPXO8 [26], but updated for the high-resolution bathymetry and coastlines.

### 4.2.2 POSYDON-POINT Snapshots

These simulations provide hourly snapshot training inputs and the subsequent truth comparison snapshots that we use in the following discussions. POSYDON-POINT experiment simulations generate a variety of variables including but not limited to  $u$ -velocity,  $v$ -velocity, salinity, and temperature in a 3-dimensional framework. In the following examples, we utilize sea surface temperature (SST) due to the ease of displaying the 2-dimensional results. It should be noted that DMD and data assimilation are easily applied in 3-dimensional cases as well.

Fig. 4-11 shows an example snapshot of the Middle Atlantic-New York Bight region. The snapshot example is of the sea surface temperature field (top plot left), sea surface temperature field zoomed (top plot right), and two sensors (bottom plot) for POSYDON-POINT Experiment simulation in the Middle Atlantic-New York Bight region on 27 August 2018. Sensors ( $s_1$  and  $s_2$ ) are as labeled and represent the sea surface temperature sensing locations from 23 August 2018 to 27 August 2018 for demonstration purposes. In order to show more detail and to show a realistic operating region for an autonomous platform, we choose a sub-region of the entire Middle Atlantic-New York Bight region as our operating and prediction area. In the following we are extracting dynamics, predicting, and assimilating in the sub-region (unless otherwise indicated). The 'truth' is arbitrarily chosen as the same ensemble member of the high fidelity POSYDON-POINT simulation for all times.

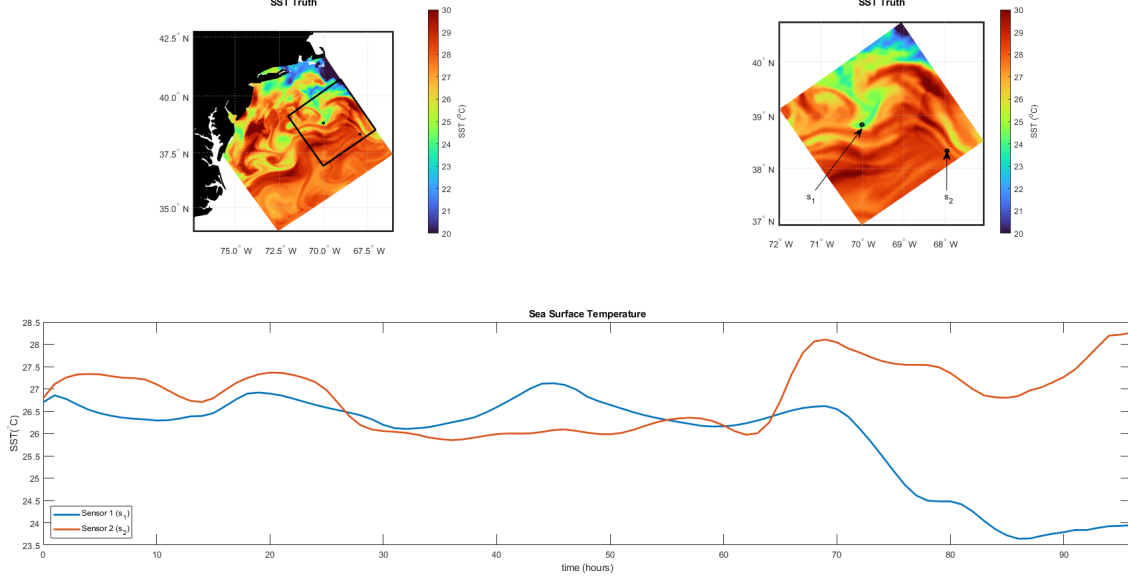


Figure 4-11: Example of sea surface temperature field (top plot left), sea surface temperature field sub-region (top plot right), and two sensors (bottom plot) for POSYDON-POINT Experiment simulation in the Middle Atlantic-New York Bight region on 27 August 2018 [65]. Sensors ( $s_1$  and  $s_2$ ) are as labeled and represent sea surface temperature sensing locations from 23 August 2018 to 27 August 2018 for demonstration purposes.

### 4.2.3 Ensemble POD Modes

In Fig. 4-12, we show the full region SST, the sub-region SST, the mean SST field of the training snapshots, the first 50 singular values, and the first four dominant POD modes of the ensemble. We once again show that although the SVD of the ensemble is a separation of variables resulting from spatial-temporal decomposition, it is still able to approximate the dominant structures required for this complex ocean simulation. The POD modes give us an idea of the extracted dominant spatial figures captured by SVD. Interestingly, around 80% of structure explaining the input snapshots is contained in just the first 4 POD modes shown. In our case we use more modes because we desire a high accuracy result for testing purposes but this just reinforces that even with a limited number of modes, results may be obtained that could be acceptable depending upon the application.

### 4.2.4 DMD of the Ensemble results for POSYDON-POINT Inputs

To apply DMD to the ensemble for this realistic ocean simulation test case we use the same training inputs that we described above in the sub-region indicated in Fig. 4-11. Like we did for the benchmark flow behind an island/cylinder test case, we extracted the DMD

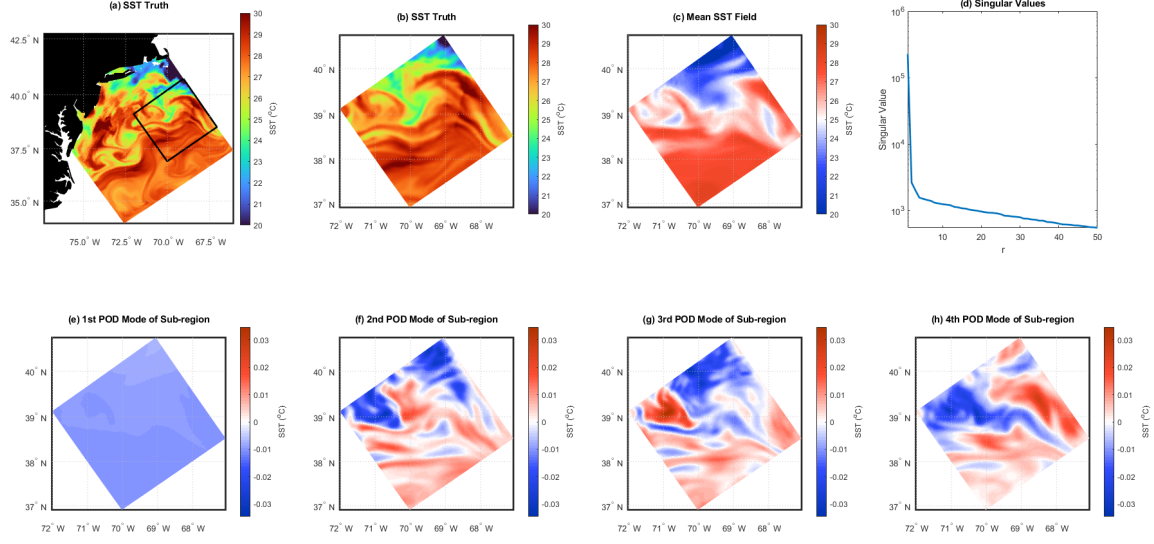


Figure 4-12: (a) Full region SST field 27 August 2018 00Z, (b) sub-region SST field 27 August 22018 00Z, (c) Mean SST field of the training snapshots, (d) POD singular values for indicated sub-region from the POSYDON-POINT Experiment simulation in the Middle Atlantic-New York Bight region for 23 Aug 2018 to 27 August 2018. The first 4 POD modes of the sub-region are shown in (e)-(h).

modes in this sub-region and used these modes along with initial conditions to determine the DMD coefficients. We then utilize the DMD coefficients of each ensemble member, DMD eigenvalues of the ensemble, and DMD modes of the ensemble to forecast the future ensemble set. For visual comparison we calculated the ensemble mean forecast and compared it to the 'truth' at the specified time in the future (beyond the training inputs). We also utilized 100 observations every hour for our DMD with DA schemes.

Figures 4-13, 4-14, 4-15, 4-16, and 4-17 show (a) True SST field for the forecast, (b) true SST field for the forecast, (c) RMSEs for DMD (blue), DMD with EnKF (orange), and DMD with GMM-DO (yellow), (d) DMD SST forecast field error, (e) DMD with EnKF SST forecast field error, and (f) DMD with GMM-DO SST forecast field error for 5 *hr*, 10 *hr*, 20 *hr*, 30 *hr*, and 40 *hr* forecasts respectively. The error fields are calculated by taking the mean of the ensemble for the indicated forecast type and subtracting the 'true' field. As we can see the DMD forecast starts off with a reasonably low RMSE and error fields at short forecast times, but that the errors increase significantly with time particularly as we can see from the 40 hour forecast (fig. 4-17). DMD with EnKF tends to to much better when the number of observations are sufficient and in this test case performs fairly well. As we can see from the plots and the RMSE values, DMD with GMM-DO performs better than DMD

with EnKF and much better than DMD without DA. These results are in agreement with the arguments and findings of [109, 110, 74, 73].

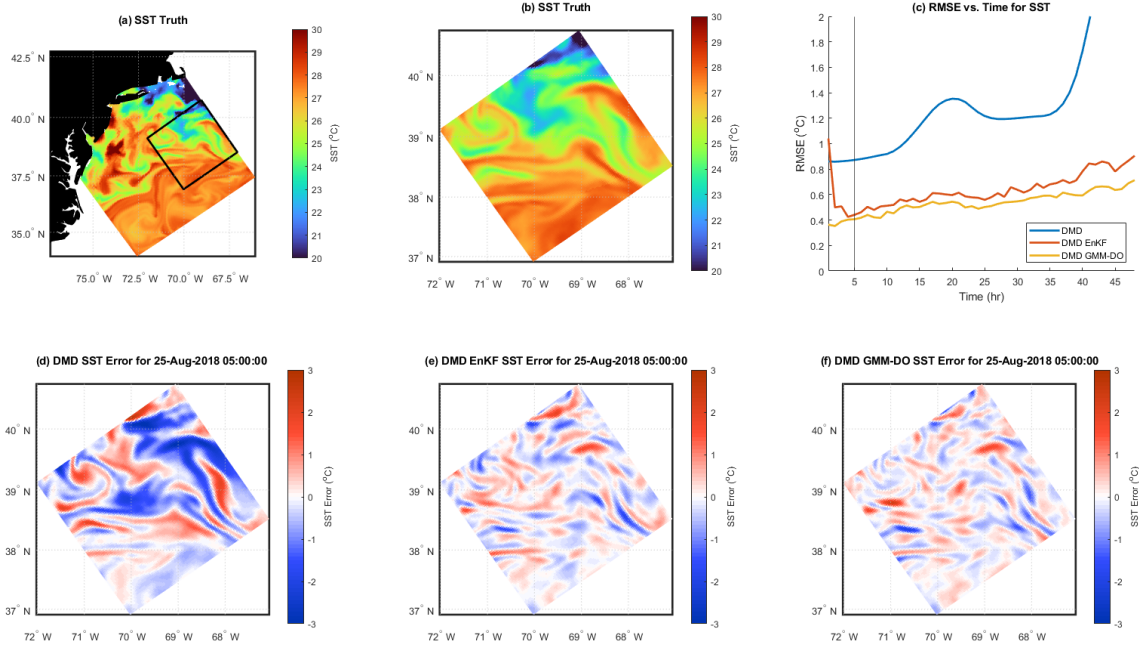


Figure 4-13: (a) True SST field for the 5 *hr* forecast, (b) true SST field for the 5 *hr* forecast, (c) RMSEs for DMD (blue), DMD with EnKF (orange), and DMD with GMM-DO (yellow), (d) DMD SST 5 *hr* forecast field error, (e) DMD with EnKF SST 5 *hr* forecast field error, and (f) DMD with GMM-DO SST 5 *hr* forecast field error.

### 4.3 Computational Costs

We include computation times for some of the methods we have discussed to exemplify feasibility for use onboard autonomous platforms. Here we were using a laptop computer with a 2.6 GHz processor clock speed. Although current small scale, easily deployed compact computers may have slightly slower clock speeds (for example the Raspberry Pi 4 has a max clock speed of 1.5 Ghz [87]), later this year they are expected to release a Raspberry Pi 5 that will likely have processing clock speeds at or above 2 GHz [88].

The most expensive step, particularly when there are large number of ensemble members and large state space, will be the SVD (which determines the POD modes) or the learned over-complete dictionary (for K-SVD). Fortunately, for the DMD models we have discussed, the SVD only needs to be computed once to train the model. This could be done on the vehicle at deployment or could be done prior to deployment on a separate computer and

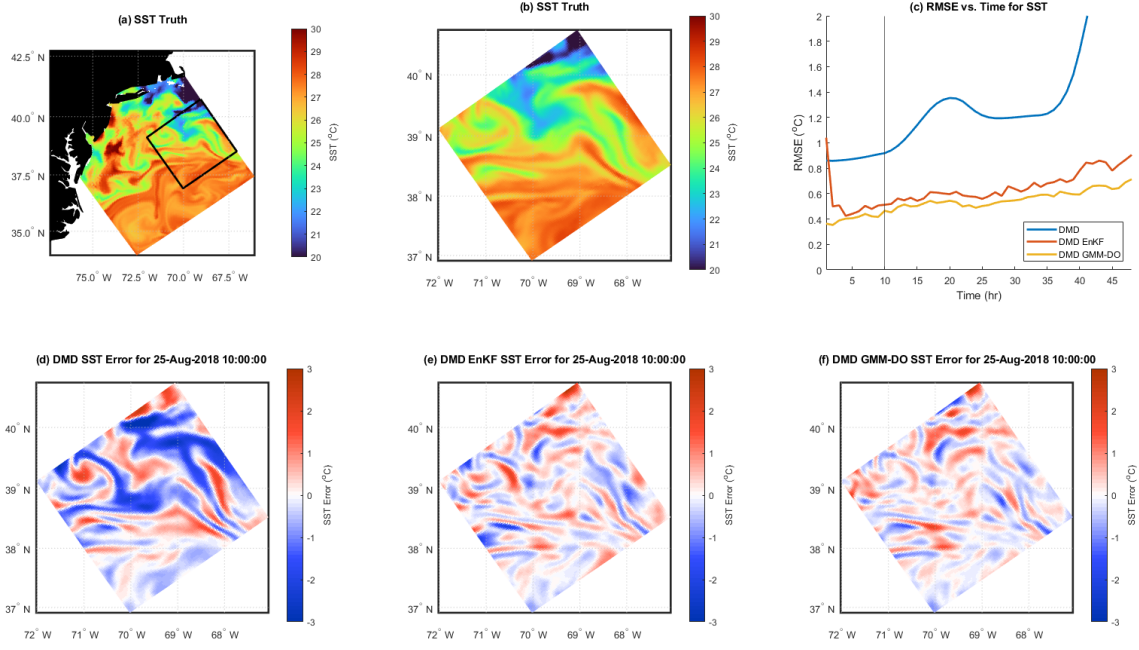


Figure 4-14: Same as Fig. 4-13 but for the 10 *hr* forecast.

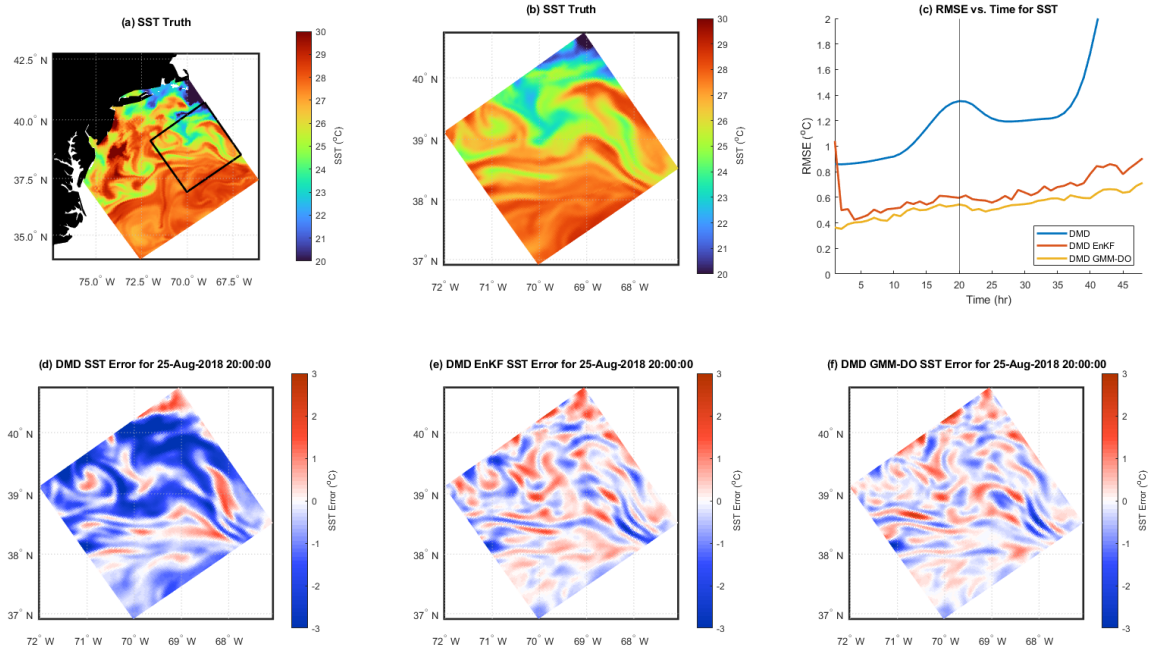


Figure 4-15: Same as Fig. 4-13 but for the 20 *hr* forecast.



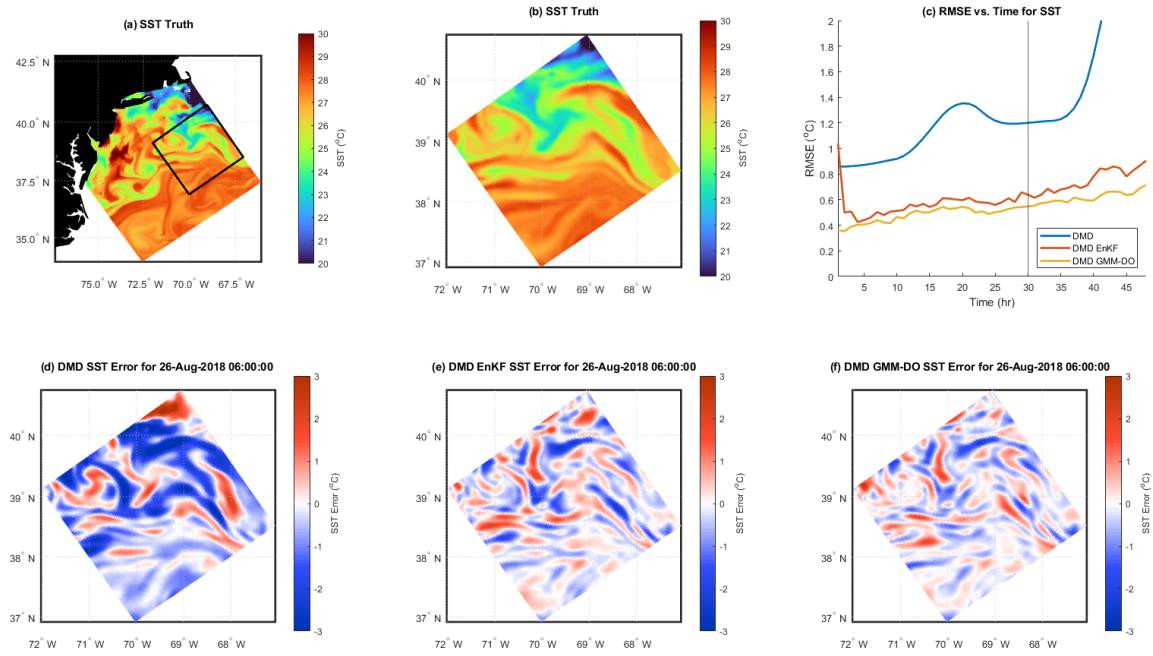


Figure 4-16: Same as Fig. 4-13 but for the 30 *hr* forecast.

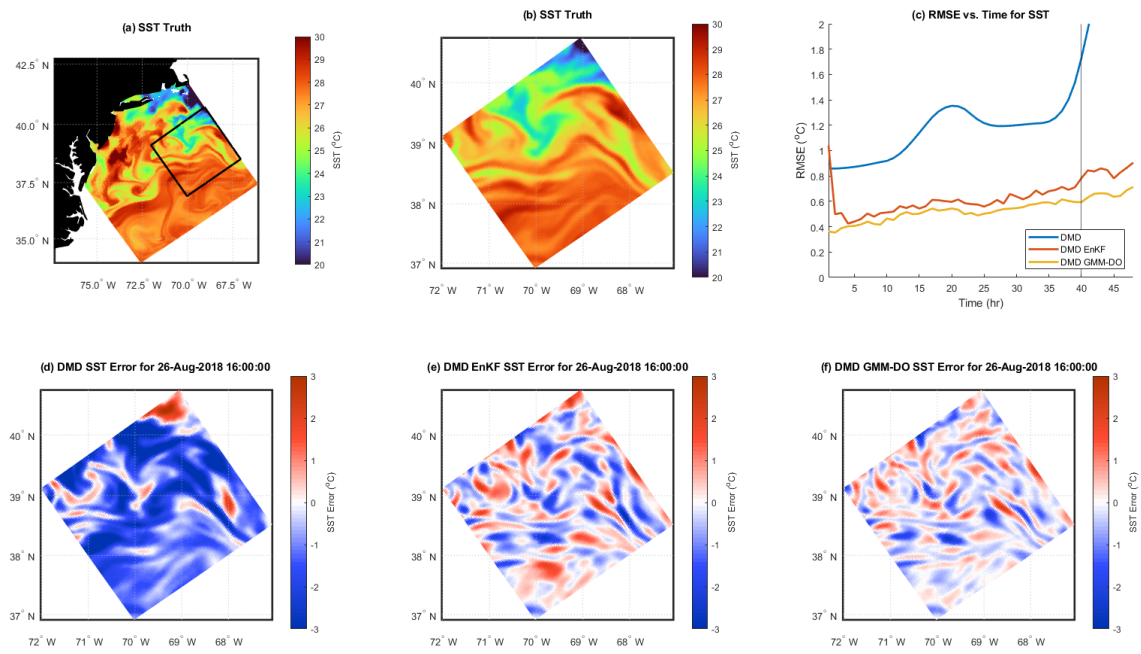


Figure 4-17: Same as Fig. 4-13 but for the 40 *hr* forecast.

the requisite POD modes, DMD modes, etc. could be pre-loaded. For all of the following times, we used the sub-region of the POSYDON POINT simulation indicated in Fig. 4-11 which has state space of size  $n = 10605$ . We use a reduced rank of  $r = 299$ , with number of ensemble members  $N = 300$ , the number of training snapshot times  $m = 49$ , and the number of observations for data assimilation of  $p = 100$ . The time to compute the compact SVD of the matrix (2.44)  $\mathbf{X} \in \mathbb{R}^{10605 \times 14700}$  and other relevant computations are shown in Table 4.1. The top several rows of the chart show computations that would only need to be made when learning the model. In the compression portion we show that compression / projection of the high-fidelity forecasts and reconstruction is relatively inexpensive. If these forecasts could be easily sent to and received by the vehicle, this option would be easy to implement and periodically could augment and improve forecasts when combined with data assimilation. Finally, the assimilation portion shows times to perform assimilation at each time  $k$  with and without DMD forecasting. Due to the linear nature of the DMD model, we also show the forecast time of advancing the GMM parameters (using  $\tilde{\mathbf{A}}$ , the  $r \times r$  projection of  $\mathbf{A}$  onto POD modes) compared to advancing the forecast in state space. In general, DMD with DA methods are efficient and utilize a relatively small amount of time and resources (once the model is learned).

Table 4.1: Computation times of indicated operation for POSYDON POINT experiment.

Operation	Time (sec)
SVD of $\mathbf{X} \in \mathbb{R}^{10605 \times 14700}$ / Compute Ensemble POD Modes	428.812
Compute $\tilde{\mathbf{A}}$ the $r \times r$ projection of $\mathbf{A}$ onto POD modes	0.568350
Compute DMD modes $\Phi$	0.687972
Compute pseudo-inverse of DMD modes $\Phi^\dagger$	0.396888
DMD forecast of ensemble from time $k$ to $k + 1$ in state space	1.523734
DMD forecast of GMM parameters from $k$ to $k + 1$ in POD subspace	0.013653
Compression	
Compression/projection of ensemble onto POD modes	1.62729
Reconstruction of ensemble compression	1.26211
Data Assimilation	
EnKF with no forecast	0.549748
DMD with EnKF from $k$ to $k + 1$	2.033897
GMM-DO (using max of mixture complexity of 5) with no forecast	5.129322
DMD with GMM-DO from $k$ to $k + 1$ (forecast in state space)	6.674947
DMD with GMM-DO from $k$ to $k + 1$ (forecast of GMM parameters in POD subspace)	5.142975



## Chapter 5

# Conclusion and Future Work

### 5.1 Conclusion

Ocean forecasting onboard autonomous platforms is inherently challenging especially when you consider the limited computing power and data storage resources available onboard these platforms. The challenges in communicating regularly with AUVs to send high fidelity forecasts (even when compressed) is one that is not likely to be completely overcome in the near future. For these reasons we have developed a method that combines DMD, an inexpensive reduction algorithm, with a Bayesian update data assimilation scheme known as GMM-DO. We explained what DMD is and showed various methods of DMD that could potentially be utilized. We also explained the GMM-DO filter and the fundamentals of its development.

We explained how we utilized our data-driven DMD with GMM-DO in a benchmark 2-dimensional flow behind an island/cylinder ensemble test case. We also utilized a more realistic ocean simulation test case from the POSYDON-POINT experiment where we showcased results for SST. We compared our technique with the same test cases but using the EnKF. Both the GMM-DO filter and EnKF were applied in the reduced stochastic subspace. In both instances, the DMD with GMM-DO outperformed DMD without DA, and DMD with EnKF in terms of accuracy.

## 5.2 Future Work

In the future we plan to continue to develop ways to implement the DMD with GMM-DO method for use onboard actual AUVs. Another goal is to develop methods to utilize DA to update the DMD models themselves based on data-model misfits and subspace augmentation [68, 69]. Finally, we plan to integrate adaptive ROM methods [101], which are used to update DMD and POD modes when significant changes to the dynamics occur, with our data assimilation for adaptive stochastic DMD. This would enable the autonomous platforms that are unable to receive forecasts computed in remote centers or platforms to make local observations that update their ROMs onboard. This could also be used to augment and update the remote center models that are making forecasts that are periodically sent to the autonomous platforms.

# Bibliography

- [1] M. Aharaon, M. Elad, and A. Bruckstein. K-svd: An algortihm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.
- [2] D. Alspach and H. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
- [3] T. Ashkam and J. N. Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, February 2018.
- [4] Roland Badeau, Bertrand David, and Gaël Richard. Fast approximated power iteration subspace tracking. *IEEE Transactions on Signal Processing*, 53(8):2931–2941, 2005.
- [5] J. G. Bellingham and K. Rajan. Robotics in remote and hostile environments. *Science*, 318(5853):1098–1102, 2007.
- [6] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 2 edition, 2008.
- [7] Manmeet S. Bhabra, Manan Doshi, Benjamin C. Koenig, Patrick J. Haley, Jr., C. Mirabito, Pierre F. J. Lermusiaux, C. A. Goudey, J. Curcio, D. Manganelli, and H. Goudey. Optimal harvesting with autonomous tow vessels for offshore macroalgae farming. In *OCEANS 2020 IEEE/MTS*, pages 1–10. IEEE, October 2020.
- [8] Michael Bianco and Peter Gerstoft. Dictionary learning of sound speed profiles. *The Journal of the Acoustical Society of America*, 141(3):1749–1758, 2017.
- [9] J. Boussinesq. *Théorie de l’écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*. Gauthier-Villars, 1897.
- [10] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations*. Wiley, 9 edition, 2008.
- [11] I. Bright, G. Lin, and J. N. Kutz. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids*, 25(12):127102, 2013.
- [12] Roger W Brockett. Dynamical systems that learn subspaces. In *Mathematical System Theory*, pages 579–592. Springer, 1991.
- [13] S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

- [14] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, June 2015.
- [15] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, 2 edition, 2001.
- [16] E. P. Chassignet, H. E. Hurlburt, O. M. Smedstad, G. R. Halliwell, P. J. Hogan, A. J. Wallcraft, R. Baraille, and R. Bleck. The hycom (hybrid coordinate ocean model) data assimilative system. *Journal of Marine Systems*, 65(1-4):60–83, 2007.
- [17] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2):215–234, 1967.
- [18] T. B. Curtin and J. G. Bellingham. Progress toward autonomous ocean sampling networks. *Deep Sea Research Part II*, 56(3):62–67, 2009.
- [19] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3):86–94, 1993.
- [20] Benoit Cushman-Roisin and Jean-Marie Beckers. *Introduction to geophysical fluid dynamics: physical and numerical aspects*. Academic press, 2011.
- [21] J. Dehaene, M. Moonen, and J. Vandewalle. *A continuous time approach to the analysis and design of parallel algorithms for subspace tracking*. Elsevier, 1995.
- [22] T. D. Dickey. Emerging ocean observations for interdisciplinary data assimilation systems. *Journal of Marine Systems*, 40:5–48, 2003.
- [23] T. D. Dickey, E. C. Itsweire, M. A. Moline, and M. J. Perry. Introduction to the limnology and oceanography special issue on autonomous and lagrangian platforms and sensors (alps). *Limnology and Oceanography*, 53, 2008.
- [24] E. O. Duda, Hart P. E., and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2001.
- [25] J. Edwards, J. Smith, A. Girard, D. Wickman, D. N. Subramani, C. S. Kulkarni, P. J. Haley, Jr., C. Mirabito, S. Jana, and P. F. J. Lermusiaux. Data-driven learning and modeling of AUV operational characteristics for optimal path planning. In *Oceans '17 MTS/IEEE Conference*, Aberdeen, June 2017.
- [26] G. D. Egbert and S. Y. Erofeeva. Efficient inverse modeling of barotropic ocean tides. *J. Atmos. Ocean. Technol.*, 19(2):183–204, 2002.
- [27] I. Eisenberger. Genesis of bimodal distributions. *Technometrics*, 6:357–363, 1964.
- [28] G. Evensen. *Data Assimilation : The Ensemble Kalman Filter*. Springer, 2nd edition, 2009.
- [29] Florian Feppon and Pierre F. J. Lermusiaux. Dynamically orthogonal numerical schemes for efficient stochastic advection and Lagrangian transport. *SIAM Review*, 60(3):595–625, 2018.

- [30] Florian Feppon and Pierre F. J. Lermusiaux. A geometric approach to dynamical model-order reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(1):510–538, 2018.
- [31] D. L. Ferris, D. N. Subramani, C. S. Kulkarni, P. J. Haley, and P. F. J. Lermusiaux. Time-optimal multi-waypoint mission planning in dynamic environments. In *OCEANS Conference 2018*, Charleston, SC, October 2018. IEEE.
- [32] National Centers for Environmental Prediction (NCEP). Ncep. <https://www.emc.ncep.noaa.gov/index.php?branch=GFS>, , August 2021.
- [33] Peter Gerstoft, Christoph F Mecklenbräuker, Woojae Seong, and Michael Bianco. Introduction to compressive sensing in acoustics, 2018.
- [34] R. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, 1991.
- [35] Michael M. Gilday. Cno navigation plan 2021, 2021.
- [36] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. In *Institute of Physics, Inverse Problems*, pages 1–26, 2002.
- [37] P. J. Haley, Jr., A. Agarwal, and P. F. J. Lermusiaux. Optimizing velocities and transports for complex coastal regions and archipelagos. *Ocean Modeling*, 89:1–28, 2015.
- [38] Patrick J. Haley, Jr. and Pierre F. J. Lermusiaux. Multiscale two-way embedding schemes for free-surface primitive equations in the “Multidisciplinary Simulation, Estimation and Assimilation System”. *Ocean Dynamics*, 60(6):1497–1537, December 2010.
- [39] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349—368, April 2017.
- [40] M. S. Hemati, M. O. Williams, and C. W. Rowley. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids*, 26(11), June 2014.
- [41] Jacob P. Heuss, Patrick J. Haley, Jr., Chris Mirabito, Emanuel Coelho, Martha C. Schönau, Kevin Heaney, and Pierre F. J. Lermusiaux. Reduced order modeling for stochastic prediction onboard autonomous platforms at sea. In *OCEANS 2020 IEEE/MTS*, pages 1–10. IEEE, October 2020.
- [42] P. Holmes, J. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry*. Cambridge University Press, 1996.
- [43] P. Jackson. *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co., 3 edition, 1998.
- [44] I Jolliffe. T.(2002) principal component analysis.
- [45] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

- [46] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT Numerical Mathematics*, 15:49–57, March 1975.
- [47] Othmar Koch and Christian Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007.
- [48] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the USA*, 17(5):315–318, 1931.
- [49] Bernard O Koopman and John V Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the USA*, 18(3):255, 1932.
- [50] Chinmay S. Kulkarni and Pierre F. J. Lermusiaux. Three-dimensional time-optimal path planning in the ocean. *Ocean Modelling*, 152, August 2020.
- [51] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dyanmic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, Philadelphia, Pennsylvania, 2016.
- [52] J Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.
- [53] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [54] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.
- [55] P. F. J. Lermusiaux. Data assimilation via Error Subspace Statistical Estimation, part II: Mid-Atlantic Bight shelfbreak front simulations, and ESSE validation. *Monthly Weather Review*, 127(7):1408–1432, July 1999.
- [56] P. F. J. Lermusiaux. Estimation and study of mesoscale variability in the Strait of Sicily. *Dynamics of Atmospheres and Oceans*, 29(2):255–303, 1999.
- [57] P. F. J. Lermusiaux. Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay. *Journal of Marine Systems*, 29(1):385–422, 2001.
- [58] P. F. J Lermusiaux. Adaptive modeling, adaptive data assimilation and adaptive sampling. *Physica D: Nonlinear Phenomena*, 230(1):172–196, 2007.
- [59] P. F. J. Lermusiaux, P. J. Haley, Jr., S. Jana, A. Gupta, C. S. Kulkarni, C. Mirabito, W. H. Ali, D. N. Subramani, A. Dutt, J. Lin, A. Shcherbina, C. Lee, and A. Gangopadhyay. Optimal planning and sampling predictions for autonomous and Lagrangian platforms and sensors in the northern Arabian Sea. *Oceanography*, 30(2):172–185, June 2017. Special issue on Autonomous and Lagrangian Platforms and Sensors (ALPS).
- [60] P. F. J. Lermusiaux, T. Lolla, P. J. Haley, Jr., K. Yigit, M. P. Ueckermann, T. Sondergaard, and W. G. Leslie. Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles. In Tom Curtin, editor, *Springer Handbook of Ocean Engineering: Autonomous Ocean Vehicles, Subsystems and Control*, chapter 21, pages 481–498. Springer, 2016.

- [61] P. F. J. Lermusiaux, D. N. Subramani, J. Lin, C. S. Kulkarni, A. Gupta, A. Dutt, T. Lolla, P. J. Haley, Jr., W. H. Ali, C. Mirabito, and S. Jana. A future for intelligent autonomous ocean observing systems. *Journal of Marine Research*, 75(6):765–813, November 2017. The Sea. Volume 17, The Science of Ocean Prediction, Part 2.
- [62] Pierre Lermusiaux. *Error subspace data assimilation methods for ocean field estimation: theory, validation and applications*. PhD thesis, Harvard University, Cambridge, Massachusetts, September 1997.
- [63] Pierre F. J. Lermusiaux. Numerical fluid mechanics. MIT OpenCourseWare, May 2015.
- [64] Pierre F. J. Lermusiaux, P. Malanotte-Rizzoli, D. Stammer, J. Carton, J. Cummings, and A. M. Moore. Progress and prospects of U.S. data assimilation in ocean research. *Oceanography*, 19(1):172–183, 2006.
- [65] Pierre F. J. Lermusiaux, Chris Mirabito, Patrick J. Haley, Jr., Wael Hajj Ali, Abhinav Gupta, Sudip Jana, Eugene Dorfman, Alison Laferriere, Aaron Kofford, G. Shepard, M. Goldsmith, Kevin Heaney, Emanuel Coelho, J. Boyle, J. Murray, L. Freitag, and A. Morozov. Real-time probabilistic coupled ocean physics-acoustics forecasting and data assimilation for underwater GPS. In *OCEANS 2020 IEEE/MTS*, pages 1–9. IEEE, October 2020.
- [66] Pierre F. J. Lermusiaux and A. R. Robinson. Data assimilation via Error Subspace Statistical Estimation, part I: Theory and schemes. *Monthly Weather Review*, 127(7):1385–1407, 1999.
- [67] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [68] Jing Lin. *Bayesian Learning for High-Dimensional Nonlinear Systems: Methodologies, Numerics and Applications to Fluid Flows*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, September 2020.
- [69] Jing Lin and Pierre F. J. Lermusiaux. Minimum-correction second-moment matching: Theory, algorithms and applications. *Numerische Mathematik*, 147(3):611–650, March 2021.
- [70] Sri Venkata Tapovan Lolla. *Path Planning and Adaptive Sampling in the Coastal Ocean*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, February 2016.
- [71] T. Lolla, P. J. Haley, Jr., and P. F. J. Lermusiaux. Time-optimal path planning in dynamic flows using level set equations: Realistic applications. *Ocean Dynamics*, 64(10):1399–1417, 2014.
- [72] T. Lolla, P. J. Haley, Jr., and P. F. J. Lermusiaux. Path planning in multiscale ocean flows: Coordination and dynamic obstacles. *Ocean Modelling*, 94:46–66, 2015.
- [73] T. Lolla and P. F. J. Lermusiaux. A Gaussian mixture model smoother for continuous nonlinear stochastic dynamical systems: Applications. *Monthly Weather Review*, 145:2763–2790, July 2017.

- [74] T. Lolla and P. F. J. Lermusiaux. A Gaussian mixture model smoother for continuous nonlinear stochastic dynamical systems: Theory and scheme. *Monthly Weather Review*, 145:2743–2761, July 2017.
- [75] T. Lolla and P. F. J. Lermusiaux. A forward reachability equation for minimum-time path planning in strong dynamic flows. *SIAM Journal on Control and Optimization*, 2020. Sub-judice.
- [76] T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, Jr. Time-optimal path planning in dynamic flows using level set equations: Theory and schemes. *Ocean Dynamics*, 64(10):1373–1397, 2014.
- [77] Tapovan Lolla, Mattheus P. Ueckermann, Konur Yiğit, Patrick J. Haley, Jr., and Pierre F. J. Lermusiaux. Path planning in time dependent flow fields using level set methods. In *IEEE International Conference on Robotics and Automation (ICRA)*, 14-18 May 2012, pages 166–173, 2012.
- [78] Peter Lu and Pierre F. J. Lermusiaux. Bayesian learning of stochastic dynamical models. *Physica D*, 2021. Sub-judice.
- [79] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [80] Jan Mandel. Efficient implementation of the ensemble kalman filter. *Center for Computational Mathematics Reports: University of Colorado at Denver and Health Sciences Center*, 231, 2006.
- [81] G. J. McLachlan and K. E. Basfor. *Mixture Models: Inference and applications to clustering*. Macel Dekker, Inc., 1988.
- [82] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley-Interscience, 1997.
- [83] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley-Interscience, 2000.
- [84] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- [85] Igor Mezić and Andrzej Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1-2):101–133, 2004.
- [86] C. Mirabito, D. N. Subramani, T. Lolla, P. J. Haley, Jr., A. Jain, P. F. J. Lermusiaux, C. Li, D. K. P. Yue, Y. Liu, F. S. Hover, N. Pulsone, J. Edwards, K. E. Railey, and G. Shaw. Autonomy for surface ship interception. In *Oceans '17 MTS/IEEE Conference*, Aberdeen, June 2017.
- [87] Vishnu Mohan. How to safely overclock raspberry pi 4 (a comprehensive guide). <https://raspberrypexpert.com/overclock-raspberry-pi-4/>, May 2021.
- [88] Vishnu Mohan. Raspberry pi 5: Release date, specs, price & rumors. <https://raspberrypexpert.com/raspberry-pi-5-release-date-specs-price/#:~:text=%20Raspberry%20Pi%20%25%20Specifications%20%28Expected%29%20%201,7%205V%20DC%20via%20GPIO%20header%20More%20/>, March 2021.



- [89] MSEAS Group. MSEAS Software, 2013.
- [90] Antonio Navarra and Valeria Simoncini. *A guide to empirical orthogonal functions for climate data analysis*. Springer Science & Business Media, 2010.
- [91] J. W. Nicholson and A. J. Healey. The present state of autonomous vehicle (auv) applications and technologies. *Marine Technology Society Journal*, 42(1):44–51, 2008.
- [92] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [93] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.
- [94] Benjamin Peherstorfer and Karen Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [95] Benjamin Peherstorfer and Karen Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.
- [96] S. R. Ramp, R. E. Davis, N. E. Leonard, I. Shulman, Y. Chao, A. R. Robinson, J. Marsden, P. F. J. Lermusiaux, D. M. Fratantoni, J. D. Paduan, F. P. Chavez, F. L. Bahr, S. Liang, W. Leslie, and Z. Li. Preparing to predict: The second Autonomous Ocean Sampling Network (AOSN-II) experiment in the Monterey Bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3–5):68–86, February 2009.
- [97] Matthew Reed. An interview with jack crenshaw. <http://www.trs-80.org/interview-jack-crenshaw/>, February 2009.
- [98] Allan R. Robinson and Pierre F. J. Lermusiaux. Data assimilation for modeling and predicting coupled physical–biological interactions in the sea. In Allan R. Robinson, James J. McCarthy, and Brian J. Rothschild, editors, *Biological-Physical Interactions in the Sea*, volume 12 of *The Sea*, chapter 12, pages 475–536. John Wiley and Sons, New York, 2002.
- [99] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry. Underwater gliders for ocean research. *Marine Technology Society Journal*, 38:73–84, 2004.
- [100] D. L. Rudnick and Perry M. J. Alps: Autonomous and lagrangian platforms and sensors. *Report of the Workshop held March 31-April 2, 2003*, 2003.
- [101] Tony Ryu, Jacob P. Heuss, Patrick J. Haley, Jr., Chris Mirabito, Emanuel Coelho, Martha C. Schönau, Kevin Heaney, and Pierre F. J. Lermusiaux. Adaptive stochastic reduced order modeling for autonomous ocean platforms. In *OCEANS 2021 IEEE/MTS*, pages 1–2. IEEE, September 2021. In press.
- [102] T. P. Sapsis. *Dynamically Orthogonal Field Equations for Stochastic Fluid Flows and Particle Dynamics*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, February 2011.

- [103] Themistoklis P. Sapsis and Pierre F. J. Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23–24):2347–2360, December 2009.
- [104] Themistoklis P. Sapsis and Pierre F. J. Lermusiaux. Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty. *Physica D: Nonlinear Phenomena*, 241(1):60–76, 2012.
- [105] O. Schofield, S. Glenn, J. Orcutt, M. Arrott, M. Meisinger, A. Gangopadhyay, W. Brown, R. Signell, M. Moline, Y. Chao, S. Chien, D. Thompson, A. Balasuriya, P. F. J. Lermusiaux, and M. Oliver. Automated sensor networks to advance ocean science. *Eos Trans. AGU*, 91(39):345–346, September 2010.
- [106] D. Shah. Algorithms for inference. MIT OpenCourseWare, December 2014.
- [107] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1992.
- [108] T. Sondergaard. Data Assimilation with Gaussian Mixture Models using the Dynamically Orthogonal Field Equation. Master’s thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, September 2011.
- [109] T. Sondergaard and P. F. J. Lermusiaux. Data assimilation with Gaussian Mixture Models using the Dynamically Orthogonal field equations. Part I: Theory and scheme. *Monthly Weather Review*, 141(6):1737–1760, 2013.
- [110] T. Sondergaard and P. F. J. Lermusiaux. Data assimilation with Gaussian Mixture Models using the Dynamically Orthogonal field equations. Part II: Applications. *Monthly Weather Review*, 141(6):1761–1785, 2013.
- [111] H. Stommel. The sloccum mission. *Oceanography*, 2(1):22–25, 1989.
- [112] D. N. Subramani, P. J. Haley, Jr., and P. F. J. Lermusiaux. Energy-optimal path planning in the coastal ocean. *Journal of Geophysical Research: Oceans*, 122:3981–4003, 2017.
- [113] D. N. Subramani and P. F. J. Lermusiaux. Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization. *Ocean Modeling*, 100:57–77, 2016.
- [114] D. N. Subramani, Q. J. Wei, and P. F. J. Lermusiaux. Stochastic time-optimal path-planning in uncertain, strong, and dynamic flows. *Computer Methods in Applied Mechanics and Engineering*, 333:218–237, 2018.
- [115] Deepak N. Subramani and Pierre F. J. Lermusiaux. Risk-optimal path planning in stochastic dynamic environments. *Computer Methods in Applied Mechanics and Engineering*, 353:391–415, August 2019.
- [116] Deepak Narayanan Subramani. *Probabilistic Regional Ocean Predictions: Stochastic Fields and Optimal Planning*. PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, February 2018.

- [117] L. Sirovich. Turbulence and the dynamics of coherent structures parts i-iii. *Quarterly of Applied Mathematics*, 45(3):561–571, October 1987.
- [118] K. Taira, S. L. Brunton, D. M. Scott, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55(12):4013–4041, December 2017.
- [119] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, January 2014.
- [120] M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016.
- [121] M. P. Ueckermann and P. F. J. Lermusiaux. 2.29 Finite Volume MATLAB Framework Documentation. MSEAS Report 14, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 2012.
- [122] B. Van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *J. Comput. Phys.*, 23(3):276–299, 1977.
- [123] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, August 1962.
- [124] Z. Zebib. Stability of viscous flow past a circular cylinder. *Journal of Engineering Mathematics*, 21:155–167, 1987.